

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-232103

(43)Date of publication of application : 27.08.1999

(51)Int.Cl.

G06F 9/44

G06F 9/06

(21)Application number : 10-030084

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 12.02.1998

(72)Inventor : FUJITA MITSUKO

INOUE SHINJI

HAYAMA SATORU

WAKE HIROYUKI

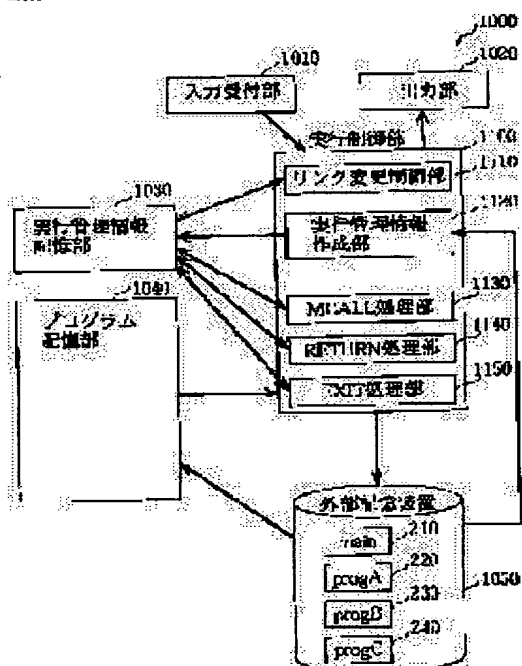
ISHIKAWA AKIRA

(54) PROGRAM LINK SYSTEM AND METHOD SEARCH SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a program link system or the like which enables fine control according to the change of balance between a request of effective use of a memory and a request of quick execution of a program.

SOLUTION: Information which is stored in an execution management information storage part 1030 and indicates whether unload of each of plural program modules constituting the program is to be executed at the end of execution of the pertinent program module or after the end of the whole of the pertinent program can be changed by operator's designation or the like which is accepted through an input acceptance part 1010 during execution of a program loaded to a program storage part 1040, and RETURN processing part 1140 which is started by interpretation execution of a RETURN code which is included in the program and means the end of execution of a program module discriminates whether unload should be executed or not based on said information.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-232103

(43) 公開日 平成11年(1999) 8月27日

(51) Int.Cl. ^a	識別記号	F I
G 0 6 F 9/44	5 3 0	G 0 6 F 9/44
9/06	4 1 0	9/06

審査請求 未請求 請求項の数17 O L (全 26 頁)

(21) 出願番号 特願平10-30084

(22) 出願日 平成10年(1998) 2月12日

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 藤田 光子

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 井上 信治

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72) 発明者 薬山 悟

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(74) 代理人 弁理士 中島 司朗

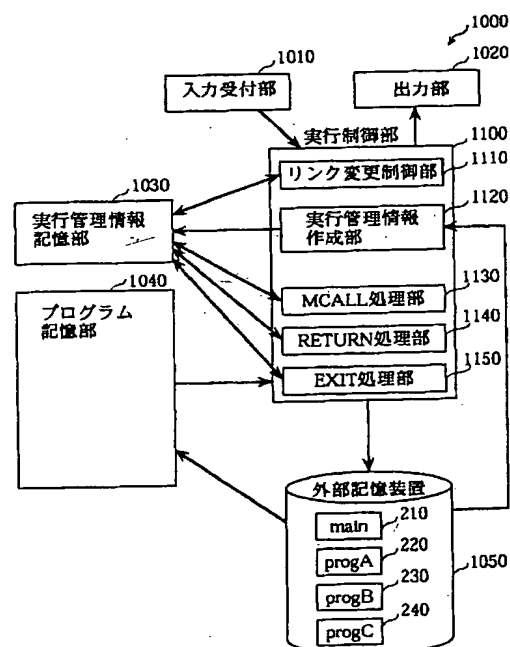
最終頁に続く

(54) 【発明の名称】 プログラムリンク方式及びメソッドサーチ方式

(57) 【要約】

【課題】 メモリの有効利用の要求と、プログラムの高速実行の要求とのバランスの変化に応じたきめ細かな制御を可能にするプログラムリンク方式等を提供する。

【解決手段】 実行管理情報記憶部1030に格納されている情報であって、プログラムを構成する複数のプログラムモジュールそれぞれについてのアンロードを当該プログラムモジュール実行終了時に行うか又は当該プログラム全体が終了するまで行わないかを示す情報を、プログラム記憶部1040にロードされたプログラムの実行中において入力受付部1010を介して受け付ける操作者の指定等により変更可能とし、前記プログラム中に含まれプログラムモジュールの実行終了を意味するRETURNコードの解釈実行により起動されるRETURN処理部1140は、前記情報に基づいてアンロードすべきか否かを判断する。



【特許請求の範囲】

【請求項1】 メモリを含み、複数のプログラムモジュールで構成されるプログラムの実行過程において前記プログラムモジュールそれぞれの前記メモリへのロード及び前記メモリからのアンロードを実行制御するプログラムリンク方式であって、

前記メモリは、複数の前記プログラムモジュールそれぞれについて当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードをするか否かを示す即時アンロード可否情報を含むプログラム管理情報を格納しており、前記プログラムモジュールを格納するための領域を有し、

複数の前記プログラムモジュールを格納する外部記憶装置と、

前記プログラムモジュールを外部記憶装置から前記メモリにロードするロード手段と、

前記プログラムモジュールを前記メモリからアンロードするためのアンロード手段と、

前記メモリに格納されたプログラム中の各コードを解釈して実行するプログラム実行制御手段とを備え、

前記プログラム実行制御手段は、プログラムモジュールの実行終了を示すコードを解釈すると当該プログラムモジュールに対応する前記即時アンロード可否情報を参照して、当該プログラムモジュールをアンロードするか否かを判断し、前記判断に基づき前記アンロード手段を起動するプログラムモジュール実行終了処理部を有することを特徴とするプログラムリンク方式。

【請求項2】 前記プログラムリンク方式はさらに、前記メモリに格納されている前記即時アンロード可否情報を前記プログラムの実行過程において変更する即時アンロード可否情報変更手段を備えることを特徴とする請求項1記載のプログラムリンク方式。

【請求項3】 前記プログラムリンク方式はさらに、操作者による前記即時アンロード可否情報の変更の指示を受け付ける入力受付手段を備え、

前記即時アンロード可否情報変更手段は前記入力受付手段の受け付けた指示に基づいて前記アンロード可否情報の変更を行うことを特徴とする請求項2記載のプログラムリンク方式。

【請求項4】 前記プログラム管理情報はさらに、前記プログラムモジュールそれぞれについて当該プログラムモジュールがプログラム実行過程において呼び出された回数を示す呼出頻度情報を含み、

前記プログラムリンク方式はさらに、

前記プログラムモジュールそれぞれがプログラム実行過程において呼び出された回数を計測して前記呼出頻度情報を更新する呼出頻度計測手段とを備えることを特徴とする請求項2又は3記載のプログラムリンク方式。

【請求項5】 前記即時アンロード可否情報変更手段は、前記プログラムモジュールそれぞれについて、前記

呼出頻度情報が所定回数未満を示す場合は当該プログラムモジュールについての前記即時アンロード可否情報を当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードする旨を示すようにし、前記呼出頻度情報が所定回数以上を示す場合は当該プログラムモジュールについての前記即時アンロード可否情報を当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードしない旨を示すようにすることを特徴とする請求項4記載のプログラムリンク方式。

【請求項6】 前記プログラムリンク方式はさらに、操作者による前記プログラム管理情報の出力指示を受け付ける第2入力受付手段と、

第2入力受付手段から前記出力指示を受け付けた旨の通知を受けたときに前記プログラム管理情報を参照してその内容を出力する出力手段とを備えることを特徴とする請求項2～5のいずれか1項に記載のプログラムリンク方式。

【請求項7】 前記即時アンロード可否情報変更手段は、あるプログラムモジュールについての前記即時アンロード可否情報が当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードしない旨を示すようにする場合において、当該プログラムモジュールが前記メモリに格納されていないときは、前記ロード手段を介して当該プログラムモジュールをロードし、前記即時アンロード可否情報が当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードする旨を示すようにする場合において、当該プログラムモジュールが前記メモリに格納されているときは、前記アンロード手段を介して当該プログラムモジュールをアンロードすることを特徴とする請求項2～6のいずれか1項に記載のプログラムリンク方式。

【請求項8】 前記外部記憶装置はさらに、これが格納する複数の前記プログラムモジュールそれぞれについての前記即時アンロード可否情報を作成するための情報である初期情報を格納し、

前記プログラムリンク方式はさらに、

あるプログラムの起動時に当該プログラムを構成する複数のプログラムモジュールそれぞれについて前記初期情報に基づき前記即時アンロード可否情報を作成して前記メモリ中へ格納をする即時アンロード可否情報作成手段を備え、

プログラム中には、プログラムの実行終了を示す1つのコードと、複数のプログラムモジュールそれぞれを呼び出す旨を示す複数のコードと、複数のプログラムモジュールそれぞれの実行終了を示す複数のコードとが含まれており、

前記プログラム実行制御手段はさらに、

プログラムの実行終了を示すコードを解釈すると当該プログラムを構成する複数のプログラムモジュールのうちメモリに格納されているものすべてを前記アンロード手

段を介してアンロードするプログラム実行終了処理部と、

プログラムモジュールを呼び出す旨を示すコードを解釈すると当該プログラムモジュールがメモリに格納されていないときには当該プログラムモジュールを前記ロード手段を介してロードするプログラムモジュール呼出処理部とを有することを特徴とする請求項1〜7のいずれか1項に記載のプログラムリンク方式。

【請求項9】 前記プログラムリンク方式はさらに、プログラムを構成する複数のプログラムモジュールそれぞれについて前記初期情報を生成して前記外部記憶装置に格納する初期情報生成手段と、前記複数のプログラムモジュールそれぞれを呼び出す旨のコードと、前記複数のプログラムモジュールそれぞれの実行終了を示すコードとを含む実行可能なコード列からなるプログラムを生成するプログラムコード生成手段と、

前記プログラムコード生成手段によって生成したプログラムを複数のプログラムモジュールに分割して、前記プログラムモジュールそれぞれを前記外部記憶装置に格納することを特徴とする請求項1〜8のいずれか1項に記載のプログラムリンク方式。

【請求項10】 複数のプログラムモジュールで構成されるプログラムの実行過程において、前記プログラムモジュールそれぞれのメモリへのロード及びメモリからのアンロードを実行制御するために用いられる情報を生成するプログラムリンク方式であって、プログラムを構成する複数のプログラムモジュールそれぞれについて当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードをするか否かを示す情報である初期情報を生成する初期情報生成手段と、

前記複数のプログラムモジュールそれぞれを呼び出す旨のコードと、前記複数のプログラムモジュールそれぞれの実行終了を示すコードとを含む実行可能なコード列からなるプログラムを生成するプログラムコード生成手段とを備えることを特徴とするプログラムリンク方式。

【請求項11】 前記初期情報生成手段は、前記プログラムを構成する複数のプログラムモジュールそれぞれについて、当該プログラムモジュールを呼び出す旨のコードの個数を合計した値が所定数以上の場合には、当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードする旨の前記初期情報を生成し、所定数未満の場合には、当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードしない旨の前記初期情報を生成することを特徴とする請求項1〜10記載のプログラムリンク方式。

【請求項12】 動的オブジェクト指向システムにおいてあるオブジェクトからスーパーオブジェクトの有するメソッドを実行するために当該メソッドの所在を検索す

るメソッドサーチ方式であって、

複数のオブジェクトを格納し、メソッド名と当該メソッドの所在を示すリンク情報との組であるメソッド情報を複数格納し得るメモリと、

実行すべきメソッド名を受け付けるメソッド名受付手段と、

前記メソッド名受付手段により受け付けたメソッド名で示されるメソッドのリンク情報を、前記メモリ中に前記メソッド情報が格納されている場合は、前記メソッド情報から検索し、検索できないとき又は前記メモリ中に前記メソッド情報が格納されていない場合は、前記リンク情報を前記のあるオブジェクトと継承関係にあるオブジェクトのメソッドスロットから検索して取得した上、前記メソッド名と取得した前記リンク情報とをメソッド情報として前記メモリに格納するメソッドサーチ手段と、特定のメソッド情報を前記メモリから削除する削除手段とを備えることを特徴とするメソッドサーチ方式。

【請求項13】 前記メソッドサーチ方式はさらに、削除すべきメソッド情報を特定するためのメソッド情報特定情報を受け付ける削除メソッド受付手段を備え、前記削除手段は、前記削除メソッド受付手段により受け付けたメソッド情報特定情報で示されるメソッド情報を前記メモリから削除することを特徴とする請求項12記載のメソッドサーチ方式。

【請求項14】 前記メソッド情報はさらに、それが含むメソッド名に対応するメソッドになされた実行要求の回数に関する情報である頻度値を含み、前記メソッド名受付手段はさらに、受け付けたメソッド名を含むメソッド情報が前記メモリ中に存在する場合にはメソッド情報中の頻度値をカウントアップし、前記削除手段は、所定の要因により起動されるものであり、所定頻度より小さい頻度値を含むメソッド情報をすべて前記メモリから削除することを特徴とする請求項12記載のメソッドサーチ方式。

【請求項15】 複数のプログラムモジュールで構成されるプログラムの実行過程において前記プログラムモジュールそれぞれについて外部記憶装置からメモリへのロード及びメモリからのアンロードを実行制御するプログラムリンク処理を、前記メモリを備えるコンピュータに実行させるためのプログラムを記録した記録媒体であって、

前記メモリは、複数の前記プログラムモジュールそれぞれについて当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードをするか否かを示す即時アンロード可否情報を含むプログラム管理情報を格納しており、前記プログラムモジュールを格納するための領域を有し、

前記プログラムリンク処理は、前記プログラムモジュールを前記外部記憶装置から前記メモリにロードするロードステップと、

前記プログラムモジュールを前記メモリからアンロードするためのアンロードステップと、

前記メモリに格納されたプログラム中の各コードを解読して実行するプログラム実行制御ステップとを含み、

前記プログラム実行制御ステップは、プログラムモジュールの実行終了を示すコードを解読すると当該プログラムモジュールに対応する前記即時アンロード可否情報を参照して、当該プログラムモジュールをアンロードするか否かを判断し、前記判断に基づき前記アンロードステップを実行するプログラムモジュール実行終了サブステップを含むことを特徴とする記録媒体。

【請求項16】 前記プログラムリンク処理はさらに、前記メモリに格納されている前記即時アンロード可否情報を前記プログラムの実行過程において変更する即時アンロード可否情報変更ステップを含むことを特徴とする請求項15記載の記録媒体。

【請求項17】 動的オブジェクト指向システムにおいて、メモリ中に存在するあるオブジェクトからスーパーオブジェクトの有するメソッドを実行するために当該メソッドの所在を検索するメソッドサーチ処理を、前記メモリを備えるコンピュータに実行させるためのプログラムを記録した記録媒体であって、

前記メモリは、複数のオブジェクトを格納し、メソッド名と当該メソッドの所在を示すリンク情報との組であるメソッド情報を複数格納し得るものであり、

前記メソッドサーチ処理は、

実行すべきメソッド名を受け付けるメソッド名受付ステップと、

前記メソッド名受付ステップにより受け付けたメソッド名で示されるメソッドのリンク情報を、前記メモリ中に前記メソッド情報が格納されている場合は、前記メソッド情報から検索し、検索できないとき又は前記メモリ中に前記メソッド情報が格納されていない場合は、前記リンク情報を前記のあるオブジェクトと継承関係にあるオブジェクトのメソッドスロットから検索して取得した上、前記メソッド名と取得した前記リンク情報とをメソッド情報として前記メモリに格納するメソッドサーチステップと、

特定のメソッド情報を前記メモリから削除する削除ステップとを含むことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、プログラムのメモリへのロードを制御するプログラムリンク方式、及び動的オブジェクト指向システムにおいてメソッドを検索するメソッドサーチ方式に関する。

【0002】

【従来の技術】まず、従来のプログラムのリンクに関する技術について説明する。コンピュータ上で動作させるプログラムが、複数の部分的なプログラムモジュールで

構成されている場合に、それぞれのプログラムモジュールをメモリにロードして実行に必要なアドレスの解決をする制御の方式として、静的リンク方式と動的リンク方式がある。

【0003】静的リンク方式とは、プログラムモジュール間のアドレス解決を予め行っている方式であり、静的にリンクされたプログラムは、主メモリ中に一体としてロードされ実行される。また、動的リンク方式とは、コンピュータにおける有限なメモリ資源を有効活用するために、実行過程において必要となったプログラムモジュールを動的にアドレス解決して実行するものであり、一般には、必要なプログラムモジュールを必要時に外部記憶装置から主メモリにロードし、アドレス解決を行い、これを実行する方式である。

【0004】一般に動的リンク方式は、静的リンク方式と比較して、メモリ資源の有効利用を可能とするという長所を有するが、外部記憶装置へのI/Oやアドレス解決のための実行時間を必要とするため、プログラムの実行速度が遅くなるという短所を有する。この短所を克服し高速にリンクすることを目的としたプログラムモジュールの動的リンクの方式として、必要時に外部記憶装置から直接ロードするのではなく、予め仮想記憶空間に対象となるプログラムモジュールをロードしておき当該プログラムモジュールを必要とする時にアドレスを解決する方式等が考えられている。

【0005】例えば、特開昭61-160135号公報に記載されている技術がある。この方式は、プログラムモジュールの管理テーブルを使用して、プログラムモジュールのロード状態を管理するものである。即ち、実行しているプログラムからあるプログラムモジュールのリンク要求があった場合には、ローダがプログラムモジュールの管理テーブルを参照して、当該プログラムモジュールを既にメモリにロードしている場合には、そのプログラムモジュールのアドレスを返し、ロードしていない場合は、メモリにロードした後にアドレスを返すという方式である。

【0006】ここで、仮想記憶空間とは、物理的な主メモリに加えて物理的に実在する大容量の補助記憶装置を用いて、オペレーティングシステムの管理の下に実現される論理的なメモリ空間をいう。なお、仮想記憶空間を実現するためにオペレーティングシステムの仮想記憶制御としては、例えば、物理空間と論理空間の変換をページと呼ばれるブロック単位で行い、必要なページを補助記憶装置から主メモリに移す方式であるページングが行われる。ページングにおいて、主メモリのどのページを補助記憶装置に追い出すかについては、最も長い間使用されていないページを選択するLRU方式が一般に用いられる。

【0007】次に、従来の動的オブジェクト指向システムにおけるメソッドサーチの技術について説明する。動

的オブジェクト指向システムにおいては、オブジェクトが特定のメソッドを起動すべき旨のメッセージを受信したときに、当該メソッドのアドレスを求めてメソッドを起動するのであるが、メッセージを受信したオブジェクトに目的とするメソッドが定義されておらず、継承を利用して当該オブジェクトのスーパーオブジェクトの有するメソッドを起動する場合があり、この場合には、継承の上位階層にあたるスーパーオブジェクトに向けて、目的とするメソッドをサーチし、検出したメソッドを起動する。

【0008】ここで、動的オブジェクト指向システムとは、コンパイル時にメソッドやデータのアドレス解決がなされるC++等の静的なオブジェクト指向システムと異なり、実行時にメソッド等のアドレス解決を行うオブジェクト指向システムである。また、スーパーオブジェクトとは、あるオブジェクトからみて継承関係の階層構造において上位階層にあるオブジェクトをいう。なお、動的オブジェクト指向システムにおいて継承関係の階層構造は、あるオブジェクトから1つ上位の階層のオブジェクトへのポインタとして実装される。

【0009】このメソッドサーチに関し、継承の階層構造が深い場合に目的のメソッドの検索にかかる計算負荷を少なくするための方式として、特開平9-6616号公報に記載のメソッドサーチ方法がある。この方法は、各オブジェクト内にキャッシュ領域を設け、初回に起動したメソッドの名前と該当メソッドへのリンク情報をキャッシュ領域に記憶しておき、次のメソッド起動時には、キャッシュ領域を検索してキャッシュしたリンク情報を取得し、取得したリンク情報を用いてメソッドを起動する方法であり、過去に行ったメソッドサーチ時の情報を各オブジェクト内に保持しておくことにより、メソッドサーチにかかる計算負荷を少なくし、メソッドサーチを高速化することを目的とした技術である。

【0010】この技術と、前述のプログラムの動的リンクを高速化する技術とは、いずれもアドレス解決を高速化するために、一度取得した情報を保存しておき再度利用する点において共通する。

【0011】

【発明が解決しようとする課題】しかしながら、上述のようなプログラムのリンクに関する技術では、メモリの有効利用の要求とプログラムの高速実行の要求とのバランスの変化に応じたきめ細かな制御を行うことができない。例えば、それぞれのプログラムモジュールが呼び出される頻度は、プログラムの内容によって異なり、データの状況等に応じて様々に変化するものであるが、従来のプログラムのリンクに関する技術では、この変化に対応して何らかの制御を行うことができない。

【0012】また、複数のプログラムを全てロードするには主メモリの容量が不足する状況下で、動的リンク方式を用いて複数のプログラムを同時に実行している場合

において、ある時間帯においては、第1のプログラムを高速に動作させる必要があり、また別の時間帯においては第2のプログラムを高速に動作させる必要があるといった高速実行要求の変動に対しても従来のプログラムリンク技術では対応できない。

【0013】また、上述のような高速にメソッドサーチを行うための技術にも前記のプログラムリンク技術の問題と同様に、メモリの有効利用の要求とメソッドサーチの高速実行の要求とのバランスの変化に応じたきめ細かな制御を行うことができないという問題がある。即ち、従来の高速にメソッドサーチを行うための技術は、キャッシュ領域を用いることにより高速なメソッドサーチを行うものであるため、高速性よりもメモリの有効活用を重視する状況となった場合には有用ではない。

【0014】そこで、本発明はかかる問題点に鑑みてなされたものであり、有限なメモリ容量を効率よく使用するというメモリの有効利用の要求と、プログラムを高速に実行するという要求とのバランスの変化に応じたきめ細かな制御を可能にするプログラムリンク方式及びメソッドサーチ方式を提供することを目的とする。

【0015】

【課題を解決するための手段】上記課題を解決するために本発明に係るプログラムリンク方式は、メモリを含み、複数のプログラムモジュールで構成されるプログラムの実行過程において前記プログラムモジュールそれぞれの前記メモリへのロード及び前記メモリからのアンロードを実行制御するプログラムリンク方式であって、前記メモリは、複数の前記プログラムモジュールそれぞれについて当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードをするか否かを示す即時アンロード可否情報を含むプログラム管理情報を格納しており、前記プログラムモジュールを格納するための領域を有し、複数の前記プログラムモジュールを格納する外部記憶装置と、前記プログラムモジュールを外部記憶装置から前記メモリにロードするロード手段と、前記プログラムモジュールを前記メモリからアンロードするためのアンロード手段と、前記メモリに格納されたプログラム中の各コードを解読して実行するプログラム実行制御手段とを備え、前記プログラム実行制御手段は、プログラムモジュールの実行終了を示すコードを解読すると当該プログラムモジュールに対応する前記即時アンロード可否情報を参照して、当該プログラムモジュールをアンロードするか否かを判断し、前記判断に基づき前記アンロード手段を起動するプログラムモジュール実行終了処理部を有することを特徴とする。

【0016】プログラムモジュールを必要になった都度メモリにロードすることは、プログラムモジュールが格納されている外部記憶装置のアクセスタイムの影響によりプログラムの実行速度を低下させる要因となり、一方、プログラムモジュールを不要な時にもメモリにロー

ドしておく、同時に動作可能なプログラムの数が減少するので、メモリの有効活用とプログラム実行の高速性とのバランスを調整する必要があるのだが、本発明によれば、プログラム管理情報を変更することによって、それぞれのプログラムモジュールが実行終了時にメモリからアンロードされるか否かを制御することが可能となるため、メモリの有効利用を図るか、プログラム実行の高速性を図るかの調整をすることができるようになる。

【0017】例えば、それぞれのプログラムモジュールが呼び出される頻度を考慮した調整として、プログラム管理情報の変更等により、一度しか実行されない或いは実行頻度の低いプログラムモジュールについて実行終了時にアンロードするように制御でき、合理的にメモリの無駄な使用を防ぐことができる。また、主メモリの容量が不足する状況下で、動的リンク方式を用いて第1のプログラムと第2のプログラムとを同時に実行している場合において、第1のプログラムを第2のプログラムより優先的に実行させたいときには、第2のプログラムを構成するプログラムモジュールの多くを実行終了時にアンロードすることとして、第1のプログラムを構成するプログラムモジュールについては少しを実行終了時にアンロードすることとしておくことにより、これが実現できる。

【0018】また、本発明に係るメソッドサーチ方法は、動的オブジェクト指向システムにおいてあるオブジェクトからスーパーオブジェクトの有するメソッドを実行するために当該メソッドの所在を検索するメソッドサーチ方式であって、複数のオブジェクトを格納し、メソッド名と当該メソッドの所在を示すリンク情報との組であるメソッド情報を複数格納し得るメモリと、実行すべきメソッド名を受け付けるメソッド名受付手段と、前記メソッド名受付手段により受け付けたメソッド名で示されるメソッドのリンク情報を、前記メモリ中に前記メソッド情報が格納されている場合は、前記メソッド情報から検索し、検索できないとき又は前記メモリ中に前記メソッド情報が格納されていない場合は、前記リンク情報を前記のあるオブジェクトと継承関係にあるオブジェクトのメソッドスロットから検索して取得した上、前記メソッド名と取得した前記リンク情報とをメソッド情報として前記メモリに格納するメソッドサーチ手段と、特定のメソッド情報を前記メモリから削除する削除手段とを備えることを特徴とする。

【0019】これにより、メモリ有効利用の観点からの要求に応え得る動的オブジェクト指向システムが実現できる。即ち、メソッドをサーチした後に蓄積するデータの量が増加した場合等において、このデータをメモリから削除しメモリを開放することにより、メモリの有効利用を図ることができる。

【0020】

【発明の実施の形態】以下、本発明の実施の形態につい

て、図面を用いて詳細に説明する。

<実施の形態1>以下、本発明の実施の形態1におけるプログラムリンク方式について説明する。

【0021】<プログラムコード作成装置>図1は、本発明の実施の形態1におけるプログラムリンク方式を実現するためのソースプログラムの例を示す図である。ソースプログラム100は、mainプログラムからprogA、progB、progCの3つのプログラムモジュールが呼び出される構造となっている。

【0022】ソースプログラム中のcallは、プログラムモジュールの呼び出しを意味する。ソースプログラム中のstaticは、プログラム全体の実行開始前に指定するプログラムモジュールをメモリに割り付け、当該プログラムモジュールを永続的リンクモジュールとして扱うことを意味する。ここで、永続的リンクモジュールとは、一度メモリにロードされたらプログラム全体が実行終了するまで、メモリに割り付けられたまま開放されないプログラムモジュールをいう。

【0023】このプログラムが実行された場合においては、progAというプログラムモジュールは、基本的にはmainプログラムの実行開始前にメモリに割り付けられ、その後mainプログラムの終了まではメモリに割り付けられたままとなり、そのメモリは開放されず、mainプログラムの終了時点でメモリが開放される。

【0024】また、ソースプログラム中のdynamicは、指定するプログラムモジュールを必要時に動的にメモリに割り付け、当該プログラムモジュールを一時的リンクモジュールとして扱うことを意味する。ここで、一時的リンクモジュールとは、当該プログラムモジュールの実行が終了すれば、プログラム全体が実行終了していても当該プログラムモジュールをアンロードすべきモジュールをいう。なお、アンロードとは、ロードによりプログラムモジュールが割り付けられていたメモリを開放することをいい、アンロードにより開放されたメモリは新たなロードに利用可能となる。

【0025】このプログラムが実行された場合においては、progBとprogCというプログラムモジュールは、基本的にはcallが実行されて呼び出された時点で、メモリに割り付けられ、その後当該プログラムモジュールの実行終了時に割り付けられたメモリが開放される。図2は、ソースプログラム100を参照してプログラムコード作成装置によって作成される実行形式のプログラムコードのイメージを示す図である。

【0026】プログラムコード210は、メインプログラムに関するコードであり、プログラムモジュールのリンクに関する制御情報となるモジュール属性定義部211と、機械語である実行コード212とで構成される。また、プログラムコード220、230、240は、それぞれソースプログラム100のprogA、prog

1150とを有する。

【0038】リンク変更制御部1110は、実行管理情報記憶部1030に格納されているメモリ管理情報を更新することにより、所定のプログラムモジュールを永続的リンクモジュールから一時的リンクモジュールに変更したり、一時的リンクモジュールから永続的リンクモジュールに変更したりする制御を行うものである。実行管理情報作成部1120は、プログラムを起動する際に、外部起動装置にアクセスしてプログラムコードを参照し、実行管理情報を作成し、実行管理情報記憶部1030に格納するものである。

【0039】MCALL処理部1130は、実行中のプログラムから呼び出される処理であり、プログラムモジュールのロードに関する処理を行うものである。RETURN処理部1140は、実行中のプログラムから呼び出される処理であり、プログラムモジュールをアンロードする処理を行うものである。EXIT処理部1150は、プログラムの終了に際してのメモリ開放処理等を行うものである。

【0040】図6は、メモリ管理情報のイメージを示す図である。メモリ管理情報は、プログラム毎に1単位存在するものであり、図6では、mainを中心としたプログラムについての1単位のメモリ管理情報1200のみを示している。同図に示すように、メモリ管理情報は、メインプログラム名とアドレスとサイズとの組を1つと、プログラム単位を構成する複数のプログラムモジュールそれぞれについて、プログラムモジュール名とリンク属性とアドレスとサイズとの組とで構成される。

【0041】ここで、アドレスとは、プログラムをロードするために割り当てるメモリのアドレスをいう。なお、同図に示す値は、mainプログラム起動直後の状態を示しており、一時的リンクモジュールについてはまだロードされていない状態でありアドレス値とサイズは意味を持たない。図7は、CALL情報のイメージを示す図である。

【0042】CALL情報も、プログラム毎に1単位存在するものであり、図7では、mainを中心としたプログラムについての1単位のCALL情報1210のみを示している。同図に示すように、CALL情報は、プログラムモジュール名とMCALL戻りアドレスとの組で構成されており、最後尾に終端マークが付加されている。プログラムモジュール名とMCALL戻りアドレスとの組は、プログラムモジュールの呼び出し毎にCALL情報の最後尾に追加され、プログラムモジュールの実行が終了して呼び元に戻る毎に削除されるものであり、この追加、削除についてはそれぞれ後述するMCALL処理、RETURN処理により行われる。

【0043】ここで、MCALL戻りアドレスとは、プログラムの実行中にプログラムモジュールを呼び出すためにMCALLが実行されるために、呼び出されたプロ

グラムモジュールの実行終了後にはMCALLの次の命令に制御を戻さなければならないが、その戻るべきアドレスをいう。また、CALL情報の要素数はプログラムモジュールの呼び出し状況によって変動するため、その終わりを示す意味で、終端マークを付しているが、この終端マークは、0等のプログラムモジュール名ではないことを示すコードであればよい。

【0044】なお、図7は、progAというプログラムモジュールが呼び出された後の状態を示している。以下、上述の構成及びデータ構造を備えるプログラム実行制御装置1000の動作について、図2に示すmain、progA、progB、progCからなるプログラムの実行を例にして説明する。

【0045】図8は、実行制御部1100によるプログラム起動の動作を示すフローチャートである。操作者がmainプログラムの実行を指示したときに、実行制御部1100は、入力受付部1010を介して指示を受け付け、実行管理情報作成部1120によって、外部記憶装置1050からmainのプログラムコード210を読み出し、モジュール属性定義部211を参照しメモリ管理情報1200を作成して実行管理情報記憶部1030に格納する（ステップS1311）。

【0046】但し、メモリ管理情報1200中のアドレス及びサイズについては、この段階では格納しないがその他の情報についての値は格納される。次に、実行制御部1100は、プログラムコード210の実行コード212をプログラム記憶部1040にロードし、メモリ管理情報1200中のアドレス及びサイズに値を設定する（ステップS1312）。このプログラムのロードについてのメモリ割当は、通常のオペレーティングシステムにおける制御と同様であり、メモリの一領域を確保し、当該メモリ領域に外部記憶装置からmainプログラムの実行コードを読み込む。この確保したメモリ領域のアドレスとサイズが前述のメモリ管理情報1200の値となる。

【0047】mainプログラムをロードした後、実行制御部1100は、メモリ管理情報を参照してリンク属性が永続的リンクであるプログラムモジュールの実行コードをプログラム記憶部1040にロードし、メモリ管理情報1200中のアドレス及びサイズに値を設定する（ステップS1313）。この結果として、progAというプログラムモジュールのプログラムコード220がプログラム記憶部1040にロードされ、メモリ管理情報1200の各値は、図6に示す状態となる。この例では、永続的リンクモジュールはprogAのみであるので、一時的リンクモジュールであるprogBやprogCはまだロードされない。

【0048】progAのロードの後、実行制御部1100は、mainプログラムを実行する（ステップS1314）。以下、mainプログラムの実行に即してプ

B、prog Cのプログラムモジュールに対応するものである。

【0027】なお、これらのプログラムコード210、220、230、240は、それぞれ外部記憶装置に別のファイルとして記録されるものである。モジュール属性定義部211は、プログラムモジュールが永続的リンクモジュールであるか一時的リンクモジュールであるかを示す情報であり、プログラムが起動された時点でオペレーティングシステムによってメモリに読み込まれる部分である。

【0028】実行コード212や各プログラムモジュールの実行コードは、実際は機械語で記述されているが、図中ではわかりやすく文字列で表現している。MCALL、EXIT、RETURNは、いずれもオペレーティングシステムが提供するサービスを呼び出すための機械語コードであり、MCALLはプログラムモジュールに実行の制御を移す意味、EXITはプログラム全体の終了の意味、RETURNはプログラムモジュールから呼び元に制御を戻す意味である。なお、これらの機械語コードは、例えばオペレーティングシステムのサービスを呼び出すスーパーバイザコール命令(SVC命令)等を含む1以上の機械語からなる。

【0029】これらの機械語コードをプロセッサが解読した場合には、当該プロセッサは、オペレーティングシステムにおける所定のサービスを実現するためのプログラムのアドレスに制御を移す。また、オペレーティングシステムは本発明に係るプログラムリンク方式実現のためにこれらのサービスを行うが、その処理内容については後述する。

【0030】図3は、本発明の実施の形態1におけるプログラムコード作成装置の機能ブロック図である。プログラムコード作成装置300は、ソースプログラムを参照して実行形式のプログラムコードを作成するものであり、ソースプログラムを解読するソースプログラム解読部310と、その解読結果に応じてプログラムコード中のリンク制御用情報であるモジュール属性定義部を生成するモジュール属性定義生成部320と、その解読結果に応じてプログラムコード中の実行コードを生成する実行コード生成部330とを備える。

【0031】図4は、プログラムコード作成装置300の動作を示すフローチャートである。以下、図4のフローチャートに基づいてプログラムコード作成装置300がソースプログラム100を参照してプログラムコード210等を作成する手順について説明する。

【0032】まず、プログラムコード作成装置300は、ソースプログラム解読部310により、ソースプログラム100を解読し、static又はdynamicというプログラムモジュールの属性定義文に基づいて、モジュール属性定義部211を生成する(ステップS410、S420)。次に、ソースプログラムの処理

内容部分を解読し、処理内容に対応するように機械語の実行コードを生成する(ステップS430)。この処理ステップS430は、基本的に通常のコンパイラ及びリンカの翻訳・連携編集処理と同等である。但し、プログラムモジュールの呼び出し記述については、MCALLというオペレーティングシステムのサービスに制御を移すための機械語コードに翻訳し、また、プログラムのメイン処理であるmainの処理の後尾にはEXITというサービスに制御を移すための機械語コードを付加し、その他のプログラムモジュールとなる処理の後尾にはRETURNというサービスに制御を移すための機械語コードを付加し、また、それぞれのプログラムモジュールは固定的なアドレスに配置されることを前提としない再配置可能なコードとする。

【0033】<プログラム実行制御装置>以下、本発明のプログラムリンク方式を実現するためにプログラムの実行を制御するプログラム実行制御装置について説明する。図5は、本発明の実施の形態1におけるプログラム実行制御装置の機能ブロック図である。

【0034】プログラム実行制御装置1000は、入力受付部1010と、出力部1020と、実行制御部1100と、実行管理情報記憶部1030と、プログラム記憶部1040と、外部記憶装置1050とを備える。入力受付部1010は、キーボードやポインティングデバイス等の装置からなり、操作者からのプログラム起動の指示やリンク変更の指示等各種の指示を受け付け、実行制御部1100に通知する機能を有する。

【0035】出力部1020は、実行制御部の制御を受けて情報を出力するディスプレイやプリンタ等である。実行管理情報記憶部1030は、RAMの領域であり、実行管理情報を格納する。ここで、実行管理情報とは、プログラムがロードされるメモリに関する情報であるメモリ管理情報と、メモリ開放を制御するための情報でありプログラムの呼び出し状況に関する情報であるCALL情報とをいう。

【0036】プログラム記憶部1040は、RAMの領域であり、実行するためのプログラムやデータ等を格納するものであり、オペレーティングシステムの管理下であり、指定した量のメモリ確保や開放をオペレーティングシステムが実現することは通常のコンピュータシステムと同様である。外部記憶装置1050は、実行可能なプログラムコードを格納するハードディスク装置等の二次記憶装置である。図5中では、図2に示したプログラムコードを格納している様子を示している。

【0037】実行制御部1100は、複数プログラムの実行を制御しうるオペレーティングシステムにおける一機能部分であり、プログラムの実行制御及びメモリ管理に関する機能を司り、リンク変更制御部1110と、実行管理情報作成部1120と、MCALL処理部1130と、RETURN処理部1140と、EXIT処理部

ログラム実行制御装置の各処理について各説明する。図2に示すように、mainプログラムの実行コードにMCALL progAがあり、これを実行する段階で、MCALL処理部1130に制御が移される。

【0049】図9は、MCALL処理部1130の動作を示すフローチャートである。MCALL処理部1130は、CALL情報1210にprogAという呼び出すべきプログラムモジュール名とMCALL戻りアドレスとの組を追加する(ステップS1321)。このときのCALL情報1210の値は図7に示したようにな

る。
【0050】MCALL戻りアドレスは、MCALLの直前に当該アドレスを所定のレジスタに格納する機械語命令が配置される等してあり、これによりMCALL処理部1130では、MCALL戻りアドレスを取得することができるものとする。次に、MCALL処理部1130は、メモリ管理情報1200を参照し、呼び出すプログラムモジュールのリンク属性を得て(ステップS1322)、一時的リンクであれば当該プログラムモジュールをロードするが(ステップS1324)、永続的リンクならばステップS1324はスキップして、呼び出すプログラムモジュールのアドレスを所定のレジスタに格納し(ステップS1325)、当該プログラムモジュールを実行する(ステップS1326)。

【0051】即ち、永続的リンクモジュールは、プログラム全体の起動時にロードされているために新たにロードはしないが、一時的リンクモジュールは必要時にのみロードするものであるために、ステップS1324によりロードするのである。progAは、永続的リンクモジュールであるため、MCALL処理部1130は、新たにロードは行わずに、メモリ管理情報1200を参照してアドレス値を所定のレジスタに格納して、progAを実行する。なお、プログラムモジュールは再配置可能な構造であるため、所定のレジスタに自らが配置されたアドレスを得ることにより実行可能となる。

【0052】この後、progAのRETURN(図2参照)を実行する段階で、RETURN処理部1140に制御が移される。図10は、RETURN処理部1140の動作を示すフローチャートである。RETURN処理部1140は、CALL情報1210の最後尾を参照しプログラムモジュール名とMCALL戻りアドレスとを取得する(ステップS1331)。これにより、RETURN処理部1140は、どのプログラムモジュールの実行が終了したのかを知り、CALL情報から当該プログラムモジュールについての情報を削除する(ステップS1332)。この場合は、progAの実行終了を意味するRETURNであるため、progAについての情報がCALL情報から削除される。

【0053】CALL情報からprogAについての情報を削除した後、RETURN処理部1140は、メモ

リ管理情報1200を参照し、実行が終了したプログラムモジュールのリンク属性を得て(ステップS1333)、永続的リンクでなければ実行が終了したプログラムモジュールをロードしたメモリを開放し(ステップS1335)、永続的リンクであれば、ステップS1335の処理をスキップして、MCALL戻りアドレスへ制御を戻す(ステップS1336)。

【0054】即ち、永続的リンクモジュールについては、プログラム全体が終了するまでは、アンロードせず、一時的リンクモジュールについては、当該プログラムモジュールの実行が終了した時点で、割り当てているメモリを開放することになる。progAは永続的リンクモジュールであるため、RETURN処理部1140は、progAをロードしているメモリの開放は行わず、mainプログラムのMCALL progAの次のアドレスに制御を戻し、MCALL progBが実行されることになる(図2参照)。

【0055】従って、MCALL処理部1130は、progBというプログラムモジュールを呼び出す処理を行う(図9参照)。CALL情報への情報追加(ステップS1321)は、progAの場合と同様であるが、progBは一時的リンクモジュールであるため、MCALL処理部1130は、プログラムモジュールのロード処理と、ロード時に割り当てるメモリのアドレス、サイズの値をメモリ管理情報1200に設定する処理を行う(ステップS1322、S1323、S1324)。

【0056】これにより、progBというプログラムモジュールのプログラムコード230が外部記憶装置1050からプログラム記憶部1040に読み込まれる。progBのロード後、MCALL処理部1130は、progBを割り当てたメモリのアドレスを所定のレジスタに格納し、progBを実行する(ステップS1325、S1326)。

【0057】この後、progBのRETURN(図2参照)を実行する段階で、RETURN処理部1140に制御が移される。RETURN処理部1140がCALL情報を参照しプログラムモジュール名とMCALL戻りアドレスとを取得し(ステップS1331)、CALL情報から当該プログラムモジュールの情報を削除し(ステップS1332)、メモリ管理情報を参照して、当該プログラムモジュールのリンク属性を得る(ステップS1333)までは、progAのRETURNを処理する場合と同様であるが、progBは一時的リンクモジュールであるため、progBをロードしたメモリを開放する処理がなされる点で異なる(ステップS1334、S1335)。

【0058】メモリの開放は、アドレス、サイズを指定して行うものであり、通常のオペレーティングシステムにおけるメモリ開放の処理と同様である。メモリ開放の後に、MCALL戻りアドレスへ制御を戻す(ステップ

S1336)。このようにして、mainプログラムは実行され、最後にEXITが実行されることになる(図2参照)。

【0059】EXITが実行されると、EXIT処理部1150に制御が移される。図11は、EXIT処理部1150の動作を示すフローチャートである。EXIT処理部1150は、メモリ管理情報1200を参照し、メインプログラムであるmainをロードしたメモリを、アドレス、サイズ指定により開放する(ステップS1341)。

【0060】メインプログラムのメモリ開放の後、EXIT処理部1150は、メモリ管理情報1200を参照し、リンク属性が永続的リンクであるプログラムモジュールをロードしたメモリをすべて開放する(ステップS1342)。メモリの開放の後、EXIT処理部1150は、メモリ管理情報1200及びCALL情報1210を削除する(ステップS1343)。即ち、EXIT処理部1150により、実行管理情報記憶部1030に格納されていたmainをメインプログラムとするプログラム全体に関する情報はすべて削除される。

【0061】このように、プログラム実行制御装置1000は、永続的リンクモジュールは、プログラム起動時にメモリにロードしプログラム全体の実行が終了するまでメモリに割り付けているが、一時的リンクモジュールについては、当該一時的リンクモジュールの実行が必要になった時点でメモリにロードし、当該一時的リンクモジュールの実行が終了した時点で、ロードしたメモリを開放する。

【0062】従って、プログラム実行制御装置1000配下では、呼び出す頻度の低いプログラムモジュールを一時的リンクモジュールとなるように定義しておくことにより、有限なメモリを有効に活用することができる。以下、プログラム実行制御装置1000のリンク変更制御について説明する。図12は、プログラム実行制御装置1000におけるリンク変更制御動作を示すフローチャートである。

【0063】操作者は、実行中のプログラムモジュールの一覧表示要求、或いは、特定のプログラムモジュールのリンク属性の変更指示を行うことができる。リンク変更制御部1110は、入力受付部1010を介して操作者の入力を受け付け(ステップS1351)、それが実行中のプログラムモジュールの一覧表示要求であれば(ステップS1352)、実行管理情報記憶部1030に格納されている現在実行中のプログラムに関するメモリ管理情報を参照して、プログラムモジュール名及びリンク属性の一覧であるプログラムモジュールリストを出力部1020を介して出力する(ステップS1353)。なお、現在実行中のプログラムが複数あればメモリ管理情報は複数単位存在する。

【0064】図13は、プログラムモジュールリストの

イメージを示す図である。プログラムモジュールリストは、操作者がこれを参照してリンク属性を変更するための指示をするために出力されるものであり、同図は、上述のmainプログラムが実行中である状況におけるプログラムモジュールリストを示しており、progA、progB、progCについての現在のリンク属性が表示されている。

【0065】このプログラムモジュールリストを参照する等して操作者が、特定のプログラムモジュールを指定してリンク属性の変更指示を入力したならば、リンク変更制御部1110は、入力受付部1010を介して操作者の入力を受け付け(ステップS1351)、それがリンク属性の変更指示であることを判断し(ステップS1354)、変更すべきモジュール名を取得する(ステップS1355)。

【0066】ここで、操作者によりprogA(図13参照)が指定されたとする。リンク変更制御部1110は、progAのリンク属性が現在永続的リンクであるか否かをメモリ管理情報を参照して判断し、永続的リンクであったとすると、以後は一時的リンクに変更する処理を行うことになり、まずprogAが実行中か否かを判断する(ステップS1360)。プログラムモジュールが実行中か否かの判断は、実行管理情報記憶部1030中のCALL情報の最後尾に当該プログラムモジュールの名称が格納されているか否かによって行う。

【0067】プログラムモジュールが実行中でなければ、次にリンク変更制御部1110は、メモリ管理情報を参照してアドレス、サイズを指定して当該プログラムモジュールがロードされていたメモリを開放し(ステップS1361)、メモリ管理情報のリンク属性の値を一時的リンクに変更する(ステップS1362)。また、プログラムモジュールが実行中であれば、リンク変更制御部1110は、メモリ開放の処理ステップS1361はスキップして、メモリ管理情報のリンク属性の値を一時的リンクに変更する(ステップS1362)。

【0068】これにより、progAは、永続的リンクモジュールから一時的リンクモジュールに変更されることになる。また、操作者にprogB(図13参照)が指定されたとする(ステップS1351、S1354、S1355)。リンク変更制御部1110は、progBのリンク属性が現在永続的リンクであるか否かをメモリ管理情報を参照して判断し(ステップS1356)、永続的リンクでなかったとすると、以後は永続的リンクに変更する処理を行うことになり。従って、リンク変更制御部1110は、progBが実行中か否かを判断し(ステップS1357)、実行中でなければ、progBをロードし、メモリ管理情報の値としてprogBを格納したプログラム記憶部1040の領域のアドレス、サイズを設定し(ステップS1358)、リンク属性の値を永続的リンクに変更する(ステップS1359)。

また、prog Bが実行中であれば、リンク変更制御部1110は、プログラムモジュールロードの処理ステップS1358はスキップして、メモリ管理情報のリンク属性の値を永続的リンクに変更する(ステップS1359)。

【0069】上述のように、プログラム実行制御装置1000は、プログラムモジュールについてのメモリ管理情報中のリンク属性を操作者の指示に基づいて変更し、このリンク属性をプログラム中のMCALL、RETURN、EXITの処理過程において参照することにより、永続的リンクモジュールについてはプログラム全体が終了するまでメモリに割り付け、一時的リンクモジュールについては必要時以外はメモリに割り付けないようにする。

【0070】従って、操作者は、プログラムの実行状況及びメモリ資源の使用率等に応じて臨機応変にプログラムモジュールのメモリ使用形態を変更することができる。

<プログラム実行制御装置の変形例1>以下、上述のプログラム実行制御装置1000に、プログラムモジュールの呼び出された頻度を計測して、出力する呼出頻度管理機能を付加した変形例であるプログラム実行制御装置2000について説明する。

【0071】プログラム実行制御装置2000は、プログラム実行制御装置1000と基本的に同様の構成要素を備えるが、実行管理情報記憶部1030に代えて実行管理情報記憶部2030を、リンク変更制御部1110に代えてリンク変更制御部2110を、MCALL処理部1130に代えてMCALL処理部2130を備える(図5参照)。

【0072】実行管理情報記憶部2030は、基本的に実行管理情報記憶部1030と同様であるが、メモリ管理情報1200と構造の異なるメモリ管理情報2200を格納する点でのみ実行管理情報記憶部1030と異なる。図14は、実行管理情報記憶部2030に格納されたメモリ管理情報2200のイメージを示す図である。

【0073】メモリ管理情報2200がメモリ管理情報1200と異なる点は、各プログラムモジュールについての呼出頻度の情報が付加されている点である。同図に示す例では、prog Aというプログラムモジュールの呼出頻度は10であり、prog Bについては6であり、prog Cについては1である。これは、プログラム実行中の状態を示している。

【0074】MCALL処理部2130は、基本的にMCALL処理部1130と同様であるが、MCALL処理部1130におけるMCALL処理のステップS1321とステップS1322との間に、メモリ管理情報の呼出頻度値を1だけカウントアップする処理が加わったものである。これにより、メモリ管理情報中の呼出頻度の値は、プログラムモジュールが呼び出される毎に増加

することになる。

【0075】リンク変更制御部2110は、リンク変更制御部1110と同様の動作(図12)を行うが、出力するプログラムモジュールリスト(ステップS1353参照)は、メモリ管理情報中の呼出頻度を加えたものである。図15は、リンク変更制御部2110が出力部1020を介して出力するプログラムモジュールリストのイメージを示す図である。

【0076】同図に示す例では、prog Aは呼出頻度が10、prog Bは呼出頻度が2、prog Cは呼び出し頻度が1となっている。このようにプログラム実行制御装置2000は、操作者の指示に応じて、プログラムモジュールの呼出頻度に関する情報を出力するので、これを参照して操作者は適切にリンク属性変更の指示を行うことができる。

【0077】<プログラム実行制御装置の変形例2>以下、上述のプログラム実行制御装置2000に、操作者の入力を受けなくても自動的にリンク属性を変更する機能を付加した変形例であるプログラム実行制御装置3000について説明する。プログラム実行制御装置3000は、プログラム実行制御装置2000と基本的に同様の構成要素を備えるが、リンク変更制御部2110に代えてリンク変更制御部3110を備える。

【0078】リンク変更制御部3110は、リンク変更制御部2110の機能に加えて、自動リンク変更制御の機能を備えたものである。図16は、リンク変更制御部3110の自動リンク変更制御の動作を示すフローチャートである。リンク変更制御部3110は、コンピュータの備える計時機能等を用いて定期的に同図に示す自動リンク変更制御を開始し、実行管理情報記憶部2030に格納されているすべてのメモリ管理情報を参照して、複数のプログラムモジュールそれぞれについて自動的に一時的リンクモジュール又は永続的リンクモジュールに変更する。

【0079】即ち、リンク変更制御部3110は、呼出頻度の情報を参照し、所定数以上の呼出頻度で使用されているプログラムモジュールについては永続的リンクモジュールに変更し、所定数未満の呼出頻度で使用されているプログラムモジュールについては一時的リンクモジュールに変更する。以下、図14に示すメモリ管理情報の値を前提にして、自動リンク変更制御の動作を説明する。

【0080】まず、リンク変更制御部3110は、メモリ管理情報を参照し1つのプログラムモジュールの情報に着目し(ステップS3301)、呼出頻度が所定数以上か否か判断する(ステップS3302)。ここで、prog Aに着目したとし(図14参照)、所定数は5であるとすると。呼出頻度が10であるprog Aは、所定数より呼出頻度が高いため、以後永続的リンクモジュールにするための処理を行う。

【0081】リンク変更制御部3110は、リンク属性が永続的リンクであるか否かを判断し（ステップS3307）、永続的リンクでない場合には、着目しているプログラムモジュールが実行中のときは（ステップS3308）、当該プログラムモジュールをメモリにロードし（ステップS3309）てリンク属性を永続的リンクに変更し（ステップS3310）、実行中でないときは、ステップS3309をスキップして、リンク属性を永続的リンクに変更する（ステップS3310）。なお、ロードした場合には、リンク変更制御部3110は、メモリ管理情報にアドレス、サイズの値を設定する。

【0082】また、既に永続的リンクである場合には（ステップS3307）、永続的リンクに特に変更する処理はスキップする。この結果、prog Aはもともと永続的リンクモジュールであったため、リンク属性の変更はなされない。リンク変更制御部3110は、すべてのプログラムモジュールに着目して処理が完了したならば自動リンク変更制御の処理は終了するが、まだ未処理のプログラムモジュールがあれば（ステップS3311）、次のプログラムモジュールに着目して同様に処理を行う（ステップS3312、S3302）。

【0083】次に、prog Bに着目したとすると、呼出頻度は6であり所定数より呼出頻度が高いため、以後prog Bを永続的リンクモジュールにするための処理を行う。リンク変更制御部3110は、prog Bのリンク属性が永続的リンクでないため（ステップS3307）、prog Bのリンク属性を永続的リンクに変更する（ステップS3308～S3310）。なお、prog Bが実行中でなければ、メモリへのロードも行われる（ステップS3308、S3309）。

【0084】この結果として、一時的リンクモジュールであったprog Bは、永続的リンクモジュールに変更される。また、次にprog Cに着目したとすると、呼出頻度は1であり所定数より呼出頻度が低いため、以後prog Cを一時的リンクモジュールにするための処理を行う。

【0085】prog Cは、もともと一時的リンクモジュールであったため、リンク属性の変更はなされない（ステップS3303）。仮に、prog Cが永続的リンクモジュールであったとした場合には、リンク変更制御部3110は、prog Cが実行中のときは（ステップS3304）、prog Cがロードされていたメモリを開放し（ステップS3305）、リンク属性を一時的リンクに変更し（ステップS3306）、prog Cが実行中でないときはメモリ開放のステップS3305をスキップしてリンク属性を一時的リンクに変更する（ステップS3306）。

【0086】こうしてすべてのプログラムモジュールに着目し、リンク属性の変更をした後、リンク変更制御部3110の1回の自動リンク変更制御は終了する。この

ように、プログラム実行制御装置3000は、プログラムモジュールの呼出頻度に応じて自動的にプログラムモジュールのリンク属性を変更するので、プログラム実行の高速性確保とメモリ資源の有効活用とのバランスを合理的なものとすることができる。

【0087】＜プログラム実行制御装置の変形例3＞以下、上述のプログラム実行制御装置1000のリンク属性変更に関する方式を部分的に変更したプログラム実行制御装置4000について説明する。プログラム実行制御装置4000は、プログラム実行制御装置1000と基本的に同様の構成要素を備えるが、実行管理情報記憶部1030に代えて実行管理情報記憶部4030を、実行制御部1100に代えて実行制御部4100を備える（図5参照）。即ち、プログラム実行制御装置4000は、リンク変更制御部1110に代えてリンク変更制御部4110を、実行管理情報作成部1120に代えて実行管理情報作成部4120を、MCALL処理部1130に代えてMCALL処理部4130を、RETURN処理部1140に代えてRETURN処理部4140を、EXIT処理部1150に代えてEXIT処理部4150を備える。

【0088】以下、プログラム実行制御装置4000がプログラム実行制御装置1000と相違する点について順次説明する。実行管理情報記憶部4030は、基本的に実行管理情報記憶部1030と同様であるが、メモリ管理情報1200と構造の異なるメモリ管理情報4200を格納する点でのみ実行管理情報記憶部1030と異なる。

【0089】図17は、実行管理情報記憶部4030に格納されたメモリ管理情報4200のイメージを示す図である。メモリ管理情報4200がメモリ管理情報1200と異なる点は、各プログラムモジュールについてアドレス有効フラグが付加されている点である。ここで、アドレス有効フラグとは、メモリ管理情報中のアドレス値が有効であればON、無効であればOFFを示すものである。

【0090】なお、プログラム実行制御装置4000は、プログラムモジュールをメモリ中にロード済みである場合には、当該プログラムモジュールについてのアドレス有効フラグがONとなるように制御する。従って、アドレス有効フラグが既にONであるプログラムモジュールを呼び出す必要が生じた場合には、新たにロードする必要はない。

【0091】実行管理情報作成部4120は、プログラムを起動する際に、外部起動装置にアクセスしてプログラムコードを参照し、実行管理情報を作成し、実行管理情報記憶部4030に格納するものであるものであり、実行管理情報作成部1120と基本的に同様であるが、メモリ管理情報を作成するときにすべてのプログラムモジュールについてアドレス有効フラグをOFFにする点

が異なる。

【0092】以下、実行制御部4100によるプログラム起動の動作について図2に示すプログラムの実行例に基づいて説明する。図18は、実行制御部4100によるプログラム起動の動作を示すフローチャートである。操作者がmainプログラムの実行を指示したときに、実行制御部4100は、入力受付部1010を介して指示を受け付け、実行管理情報作成部4120によって、外部記憶装置1050からmainのプログラムコード210を読み出し、モジュール属性定義部211を参照しメモリ管理情報4200を作成して実行管理情報記憶部1030に格納する(ステップS4311)。

【0093】但し、メモリ管理情報4200中のアドレス及びサイズについては、この段階では格納しないがその他の情報についての値は格納される。次に、実行制御部4100は、プログラムコード210の実行コード212をプログラム記憶部1040にロードし、メモリ管理情報4200中のアドレス及びサイズに値を設定する(ステップS4312)。

【0094】次に、実行制御部4100は、mainプログラムを実行する(ステップS4313)。このように、実行制御部4100は、プログラム実行制御装置1000における実行制御部1100と異なり、リンク属性が永続的リンクのプログラムモジュールをメインプログラムの実行前にロードしない。

【0095】以下、MCALL処理部4130、RETURN処理部4140、EXIT処理部4150について説明する。図19は、MCALL処理部4130の動作を示すフローチャートである。MCALL処理部4130は、MCALL処理部1130の動作(図9参照)におけるステップS1322、ステップS1323、ステップS1324について、MCALL処理部1130と異なる動作を行う。

【0096】即ち、MCALL処理部4130は、リンク属性を参照するステップS1322に代えてアドレス有効フラグを参照するステップS4322の処理を行い、永続的リンクか判断するステップS1323に代えて、アドレス有効フラグがONか否かを判断するステップS4323の処理を行う。従って、MCALL処理部4130は、アドレス有効フラグがOFFであれば、プログラムモジュールをロードし(ステップS4324)、アドレス有効フラグがONであれば、新たにロードは行わない。但し、ロードしたときに、アドレス有効フラグをONにする。

【0097】換言すれば、プログラムモジュールの呼び出しの指示を受け付けるMCALL処理部4130は、当該プログラムモジュールが既にロードされていれば新たにロードはしないが、ロードされていなければ新たにロードするのである。図20は、RETURN処理部4140の動作を示すフローチャートである。RETURN

N処理部4140は、RETURN処理部1140の動作(図10参照)におけるステップS1335について、RETURN処理部1140と異なる動作を行う。

【0098】即ち、RETURN処理部4140は、永続的リンクでない場合に(ステップS1334)、プログラムモジュールをロードしたメモリを開放するステップS1335に代えて、プログラムモジュールをロードしたメモリを開放してアドレス有効フラグをOFFにするステップS4335の処理を行う。図21は、EXIT処理部4150の動作を示すフローチャートである。

【0099】EXIT処理部4150は、EXIT処理部1150の動作(図11参照)におけるステップS1342について、EXIT処理部1150と異なる動作を行う。即ち、EXIT処理部4150は、リンク属性が永続的リンクであるプログラムモジュールをロードしたメモリを開放するステップS1342に代えて、アドレス有効フラグがONであるプログラムモジュールをロードしたメモリを開放するステップS4342の処理を行う。

【0100】上述のようなアドレス有効フラグを用いた制御により、操作者の指示等に基づくリンク属性の変更がなされていない間においては、プログラム実行制御装置4000は、永続的リンクモジュールの呼び出しが必要な時は、当該永続的リンクモジュールを最初の1度だけメモリにロードをし、それ以後は新たにロードすることなく呼び出しを行い、一時的リンクモジュールの呼び出しが必要な時は、その都度新たにロードして呼び出す。

【0101】以下、プログラム実行制御装置4000のリンク変更制御機能を司るリンク変更制御部4110の動作について説明する。図22は、リンク変更制御部4110の動作を示すフローチャートである。リンク変更制御部4110は、リンク変更制御部1110の動作(図12参照)におけるステップS1357、ステップS1358、ステップS1360、ステップS1361を削除したのと同様の動作を行う。

【0102】即ち、リンク変更制御部4110は、操作者が変更すべきと指示したプログラムモジュールが永続的リンクモジュールであれば、これを一時的リンクモジュールに変更するために、メモリ管理情報の当該プログラムモジュールに関するリンク属性を一時的リンクとするだけであり、操作者が変更すべきと指示したプログラムモジュールが一時的リンクモジュールであれば、これを永続的リンクモジュールに変更するために、メモリ管理情報の当該プログラムモジュールに関するリンク属性を永続的リンクとするだけである。

【0103】従って、プログラム実行制御装置4000におけるプログラムモジュールについてのリンク属性の変更の効果は、当該プログラムモジュールが呼び出されてそのプログラムモジュールの実行が終了した時点での

扱いに現れ、メモリに割り付けたままとするか、割り付けているメモリを開放するかという点においてリンク属性の変更前と異なった扱いがなされる。

<実施の形態2>以下、本発明の実施の形態2におけるメソッドサーチ方式を実現する動的オブジェクト指向システムについて説明する。

【0104】図23は、動的オブジェクト指向システム5000における各オブジェクトの構造のイメージを示す図である。動的オブジェクト指向システム5000は、CPU、メモリ、入力装置、出力装置等を備える一

般的なコンピュータと、このコンピュータ上で動作するオブジェクト指向のプログラム群とからなる。
【0105】オブジェクト5100、5200、5300は継承関係を持つオブジェクトであり、Super Objectというスロットによって親となるスーパーオブジェクトを、Sub Objectというスロットによって子となるサブオブジェクトを指し示している。即ち、Object Aは、Object B及びObject Cのスーパーオブジェクトであり、Object Bは、Object Cのスーパーオブジェクトである。

【0106】ここで、スロットとは、オブジェクトの内部でデータを保持する領域をいい、コンピュータのメモリ上に割り当てられたデータを示すポインタとして実現される。なお、Object AのSuper Objectスロット及びObject CのSub Objectスロットは、空リンクであり、それぞれ、スーパーオブジェクト、サブオブジェクトが存在しないことを示している。また、動的オブジェクト指向システム5000は、Object A～Cの他にも操作者の入力受け付け等を行う様々な制御用オブジェクトを有する。

【0107】また、Object A、Object B、Object Cは、それぞれ過去に起動したメソッドについてのリンク情報を記憶するため、メソッドリスト5110、5210、5310を有する。Methodsスロットは、各オブジェクトに定義されたメソッドを管理している。

【0108】同図では、Object AはMethod Aを有し、Object BはMethod Bを有しており、これらのメソッドは、継承関係にあるObject Cから利用することができる。図24は、メソッドリストの内容のイメージを示す図である。メソッドリストは、メソッド名とメソッドへのポインタと呼出頻度との組を1セットとして、メソッドサーチしたメソッドについて格納するリストである。

【0109】このメソッドリストは、メソッド毎に1セットの情報が追加、又は削除されるものであり、例えば、メソッド毎の1セットの情報をポインタチェーンで連結したいわゆるリスト構造等のデータ構造をとる。

<メソッドリストを用いたメソッドサーチ機構>以下、メソッドリストを用いたメソッドサーチ機構について説

明する。

【0110】まず、Object Cに対してMethod Aを起動すべきメッセージが送られてきた場合を例にしてメソッドサーチの手順について説明する。但し、まだ1度もMethod Aの起動がなされていないとし、メソッドリスト5310及びメソッドリスト5210にはMethod Aに関する情報は格納されていないものとする。

【0111】(1) Object CにMethod Aを起動すべきメッセージが送られる。

(2) Object Cは、自己のMethodsスロットを検索し、Method Aを探す。

(3) Object Cには、Method Aが定義されていないので、メソッドリスト5310内を検索する。

【0112】(4) メソッドリスト5310に、Method Aの情報が格納されていないので、Super Objectスロットを介してスーパーオブジェクトであるObject BのMethodsスロットを検索する。

(5) Object BのMethodsスロットにも、Method Aが定義されていないので、メソッドリスト5210内を検索する。

【0113】(6) メソッドリスト5210にも、Method Aの情報が格納されていないので、さらにスーパーオブジェクトであるObject AのMethodsスロットを検索する。

(7) Object Aには、Method Aが定義されているので、Method Aのポインタを得て、Method Aの名称及びポインタをメソッドリスト5310に追加登録し、メソッドリスト中の呼出頻度には1を設定する。

【0114】(8) Method Aを起動する。

(9) この後、Object Cに、再度Method Aを起動すべきメッセージが送られた場合には、Object Cは、上述の手順で自己のメソッドリスト5310を検索した段階で、Method Aのポインタを得ることができるため、即時にMethod Aを起動することができる。なお、起動の都度メソッドリスト中の呼出頻度の数値を1だけカウントアップする。

【0115】次に、それぞれのオブジェクトに関してのメソッド呼出処理に着目して説明する。図25は、それぞれのオブジェクトについて行うメソッド呼出処理の動作を示すフローチャートである。オブジェクトに対してメソッド起動要求のメッセージが送られると、オブジェクトは、自己のMethodsスロットに起動すべきメソッドが格納されているか判断し(ステップS5501)、もし格納されているならば、当該メソッドを実行する(ステップS5502)。

【0116】また、自己のMethodsスロットに起動すべきメソッドが格納されていないければ、オブジェク

トは、自己のメソッドリストに当該メソッドに関する情報が登録されているかを判断し（ステップS5503）、登録されている場合は、メソッドリスト中の当該メソッドに関する呼出頻度の値を1増加し（ステップS5507）、メソッドリスト中の当該メソッドに関するリンク情報、即ち当該メソッドのポインタを用いて当該メソッドを実行する（ステップS5508）。

【0117】また、自己のメソッドリストに当該メソッドに関する情報が登録されていない場合は（ステップS5503）、オブジェクトは、Super Object スロットを介してスーパーオブジェクトについてメソッドサーチを行い、当該メソッドのリンク情報を得る（ステップS5504）。起動すべきメソッドのリンク情報を得たオブジェクトは、メソッドの名称と、メソッドへのリンク情報であるポインタとを自己のメソッドリストに登録し（ステップS5505）、呼出頻度としては1を設定し（ステップS5506）、リンク情報に基づきメソッドを実行する（ステップS5508）。

【0118】＜メソッドリスト削除機構＞以下、メソッドリストから1又は複数のメソッドに関する情報を削除する機構について説明する。動的オブジェクト指向システム5000に対し、操作者は、メソッドリストの出力指示或いはメソッドリストの削除指示をすることができる。なお、メソッドリストの削除指示については、削除すべきメソッドリストを保持するオブジェクトの指定に加えて、メソッド名又は削除基準となる呼出頻度値を指定することを要する。

【0119】ここで、メソッドリストの削除とは、メソッドリストから1セットのメソッドに関する情報を削除することをいい、削除することによりリスト構造のチェーンから1セット分のデータが取り外され、その1セットの情報分のメモリが開放される。図26は、動的オブジェクト指向システム5000におけるメソッドリスト削除制御の動作を示すフローチャートである。

【0120】動的オブジェクト指向システム5000は、操作者の入力を受け付ける制御用オブジェクトを介して、操作者からの指示を受けつけ（ステップS5511）、それがメソッドリストの出力指示であれば（ステップS5512）、存在する各オブジェクトのメソッドリストを参照し、これらのメソッドリストの内容を出力する（ステップS5513）。

【0121】図27は、メソッドリストの内容の出力イメージを示す図である。同図に示すように、オブジェクト名とメソッド名と呼出頻度との組が一覧形式でディスプレイ、プリンタ等に出力される。また、動的オブジェクト指向システム5000は、操作者からの指示がメソッド削除指示であれば（ステップS5514）、操作者により入力されたメソッド削除指示を解釈し、削除対象のメソッドリストを有するオブジェクトに対して、操作者からの指示をパラメータに含む削除メッセージを送る

（ステップS5515）。

【0122】この結果、削除メッセージを受けたオブジェクトは、削除メッセージのパラメータの内容に応じて、メソッドリストの削除動作を行うことになる。図28は、削除メッセージを受けたオブジェクトにおけるメソッドリスト削除の動作を示すフローチャートである。削除メッセージを受けたオブジェクトは、パラメータに着目し（ステップS5521）、削除メッセージがメソッド名指定の削除指示であれば（ステップS5522）、自己の有するメソッドリストから当該メソッド名のメソッドに関する情報の1セットを削除する（ステップS5523）。

【0123】また、当該オブジェクトは、削除メッセージが呼出頻度指定によるメソッドリストからのメソッドの削除指示であれば（ステップS5524）、指定された呼出頻度を取得し（ステップS5525）、自己の有するメソッドリスト中の呼出頻度が、指定された呼出頻度未満であるメソッドの情報を検索し、検索したメソッドの情報をすべて削除する（ステップS5526）。

【0124】例えば、操作者によりObject Cを対象として呼出頻度として5を指定された削除指示を受けた場合には、Object Cは、呼出頻度が5未満であるメソッドの情報をすべて削除する。Object Cの有するメソッドリストが図24に示すようであれば、呼出頻度が10であるMethod Aに関する情報は削除されないが、呼出頻度が1であるMethod Bに関する情報は削除されることになる。

【0125】このように、動的オブジェクト指向システム5000は、操作者の指示に応じて、メソッドリストを削除して、メモリの有効活用を図る機能を有する。以上、本発明に係るプログラムリンク方式及びメソッドサーチ方式について、実施形態に基づいて説明したが、本発明はこれら実施形態に限られないことは勿論である。即ち、

（1）実施の形態1では、プログラムコード作成装置300は、プログラムモジュールの後尾にRETURNというサービスに制御を移すための機械語コードを付加することとしたが、これに限定されることはなく、プログラムモジュールの最後尾でなくても、プログラムモジュール名指定等によりどのプログラムモジュールの実行終了かを判別可能とする情報を付加するならば、前記RETURNをプログラムモジュールの呼び出し元その他の部分に付加することとしてもよい。

（2）実施の形態1では、プログラムコード210、220、230、240はそれぞれ別ファイルに分割されて記憶装置に格納されることとしたが、プログラムコード210におけるモジュール属性定義部211とmainの実行コード212とは別のファイルとしてもよく、またプログラムコード210、220、230、240をまとめて1つのファイルとしてもよい。即ち、実行制

御部1100が、モジュール属性定義部211と実行コード212とプログラムコード220、230、240とをそれぞれ識別して内容に応じてアクセスできれば、ファイル格納形態はいかなる形式であってもよい。

(3) 実施の形態1では、ソースプログラム100は、C言語に似た独自言語で記述した例であり、プログラムコード作成装置300は、ソースプログラム100中の関数構造を1つのプログラムモジュールとして扱っているが、他の形式の言語等で記述されたソースプログラムであっても、プログラムモジュールを区分する情報と、それぞれのプログラムモジュールとなる部分が永続的リンクモジュールとして扱われるべきか一時的リンクモジュールとして扱われるべきかを示す情報と、プログラムモジュールの呼出命令に相当する情報とが備わっていれば、これらの情報を参照して、プログラムコード作成装置300はプログラムコードを生成することとしてもよい。また、プログラムコード作成装置は、ソースプログラム中にstaticやdynamic等の永続的リンクモジュールか一時的リンクモジュールかについての指定がなされていない場合にも、いずれかが指定されたとみなして扱うこととしてもよい。

(4) 実施の形態1では、プログラムコード作成装置300は、ソースプログラム100を入力としており、各プログラムモジュールの後尾にはRETURNというサービスに制御を移すための機械語コードを付加し、プログラムのメイン処理の後尾にはEXITというサービスに制御を移すための機械語コードを付加し、プログラムモジュールを呼び出す記述の箇所についてはMCALLというサービスに制御を移すための機械語コードを付加することとしているが、例えば一般のリンカのように、オブジェクトプログラムの集合を入力することとして、オブジェクトプログラムに既に含まれている機械語を変更することにより上述と同様のことを実現してもよい。即ち、オブジェクトプログラム中に呼び元への復帰命令が格納されている場合は、これをRETURNというサービスに制御を移すためのコードに変えるといったような変更を施すことで、最終的に、プログラムモジュールを呼び出す部分についてはMCALL、プログラムモジュールの後尾はRETURN、メインプログラムの後尾はEXITというサービスと呼び出す機械語コードにすればよい。

(5) 実施の形態1では、プログラムコード作成装置は、ソースプログラム中にstaticやdynamic等のプログラムモジュールを永続的リンクモジュールとするか一時的リンクモジュールとするかについての情報が記述されていることを前提として、これを参照することによりモジュール属性定義部211の情報を作成することとしたが、それぞれのプログラムモジュールが呼び出される頻度をプログラム全体を解析することにより予測し、その結果に応じて、呼び出される頻度が高いプ

ログラムモジュールについては永続的リンクモジュールとし、呼び出される頻度が低いプログラムモジュールについては一時的リンクモジュールとすることとして、モジュール属性定義部211の情報を作成することとしてもよい。

(6) 実施の形態1では、プログラム実行制御装置がプログラムモジュールをメモリにロードした後、ロードしたメモリアドレスをメモリ管理情報等に保持して利用しているが、このアドレスは、メモリの物理アドレスであることとしてもよいし、オペレーティングシステムにより定めた論理アドレスであることとしてもよい。

(7) 実施の形態1では、操作者によるリンク属性の変更指示は、プログラムモジュール名のみの指定としていたが(ステップS1355参照)、操作者が、プログラムモジュール名に加えて、いかなるリンク属性に変更したいかの指定をもすることができることとして、リンク変更制御部1110は、ステップS1356の判断に代えて、操作者による変更後のリンク属性の指示を参照して永続的リンクにするか一時的リンクにするかを判断することとしてもよい。

(8) 実施の形態1におけるリンク変更制御部2110は、メモリ管理情報中の呼出頻度の値を、メモリ中の所定領域にコピーした後、メモリ管理情報中の呼出頻度の値を0にする処理を定期的に行うこととしてもよい。これにより、リンク変更制御部2110の出力するプログラムモジュールリストには、最近における各プログラムモジュールの呼び出された頻度が表示されることになる。また、リンク変更制御部3110についても同様とすると、リンク変更制御部3110は、最近における各プログラムモジュールの呼び出された状況に応じて自動的に各プログラムモジュールのリンク属性を変更することとなる。

(9) 実施の形態1では、リンク変更制御部3110は自動リンク変更制御を定期的に行うこととしたが、自動リンク変更制御の開始タイミングは操作者に指定させることとしてもよい。また、永続的リンクモジュールとするか否かの判断基準である所定数(ステップS3302参照)についてを操作者に指定させることとしてもよい。

(10) 実施の形態1では、リンク変更制御部1110は、プログラムモジュールのリンク属性を永続的リンクモジュールに変更する際に、当該プログラムモジュールがロードされていない場合はロードし(ステップS1358)、一時的リンクモジュールに変更する際に、当該プログラムモジュールがアンロードされていない場合はアンロードすることとしていたが(ステップS1361)、リンク変更制御部4110では、プログラムモジュールのリンク属性の変更の際にロード或いはアンロードはしないこととしていた(図2参照)。しかし、リンク変更制御部4110についても、リンク変更制御部1

110のステップS1358又はステップS1361のいずれか一方と同等な処理を行うこととしてもよい。

(11)実施の形態2では、動的オブジェクト指向システム5000は、操作者から削除対象のメソッドリストを有するオブジェクトの指定を受け付けることとしていたが(図26参照)、操作者により呼出頻度指定はされたがオブジェクトの指定がなされなかった場合に、すべてのオブジェクトに呼出頻度指定の削除メッセージを送ることとしてもよい。また、動的オブジェクト指向システム5000は、操作者の指示を受けつけなくても定期的に所定数の頻度指定をした削除メッセージをすべてのオブジェクトに送ることとしてもよい。また、動的オブジェクト指向システム5000は、メモリ使用率を監視する機構を備えて、メモリ使用率が所定割合より高くなったことを契機に、前記の頻度指定をした削除メッセージをすべてのオブジェクトに送ることとしてもよい。ここで、頻度指定における前記所定数は、メモリ使用率に応じて適当な値を動的に決定することとしてもよい。

(12)実施の形態1におけるプログラムリンク方式の処理手順(図4、図8～図12、図16、図18～図22のフローチャートの手順等)又は実施の形態2におけるメソッドサーチ方式の処理手順(図25、図26、図28のフローチャートの手順等)を機械語プログラムにより実現し、これを記録媒体に記録して流通・販売の対象にしても良い。このような記録媒体には、ICカード、光ディスク、フレキシブルディスク、ROM等があるが、これらに記録された機械語プログラムは汎用のハードウェアにインストールされることにより利用に供される。ここでいう汎用ハードウェアは、一般のパーソナルコンピュータ等であり、インストールした上記機械語プログラムを逐次実行して、実施の形態1に示したプログラムリンク方式又は実施の形態2に示したメソッドサーチ方式の諸機能を実現するものである。

【0126】

【発明の効果】以上の説明から明らかなように、本発明に係るプログラムリンク方式は、メモリを含み、複数のプログラムモジュールで構成されるプログラムの実行過程において前記プログラムモジュールそれぞれの前記メモリへのロード及び前記メモリからのアンロードを実行制御するプログラムリンク方式であって、前記メモリは、複数の前記プログラムモジュールそれぞれについて当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードをするか否かを示す即時アンロード可否情報を含むプログラム管理情報を格納しており、前記プログラムモジュールを格納するための領域を有し、複数の前記プログラムモジュールを格納する外部記憶装置と、前記プログラムモジュールを外部記憶装置から前記メモリにロードするロード手段と、前記プログラムモジュールを前記メモリからアンロードするためのアンロード手段と、前記メモリに格納されたプログラム

中の各コードを解読して実行するプログラム実行制御手段とを備え、前記プログラム実行制御手段は、プログラムモジュールの実行終了を示すコードを解読すると当該プログラムモジュールに対応する前記即時アンロード可否情報を参照して、当該プログラムモジュールをアンロードするか否かを判断し、前記判断に基づき前記アンロード手段を起動するプログラムモジュール実行終了処理部を有することを特徴とする。

【0127】これにより、プログラム管理情報を変更することによって、それぞれのプログラムモジュールが実行終了時にメモリからアンロードされるか否かを制御することが可能となるため、メモリの有効利用を図るか、プログラム実行の高速性を図るかの調整をすることができるようになる。ここで、前記プログラムリンク方式はさらに、前記メモリに格納されている前記即時アンロード可否情報を前記プログラムの実行過程において変更する即時アンロード可否情報変更手段を備えることとすることもできる。

【0128】これにより、プログラム実行中にプログラム管理情報を変更する機構が実現されるため、その時々状況に応じた制御をすることができるようになる。また、前記プログラムリンク方式はさらに、操作者による前記即時アンロード可否情報の変更の指示を受け付ける入力受付手段を備え、前記即時アンロード可否情報変更手段は前記入力受付手段の受け付けた指示に基づいて前記アンロード可否情報の変更を行うこととすることもできる。

【0129】これにより、操作者の意図どおりに、メモリの有効利用とプログラムの高速実行の要求のバランスをタイムリーに調整することができるようになる。即ち、プログラムはデータ次第で実行ルートが異なり、頻繁に呼び出されるプログラムモジュールがいずれであるかは変化するため、データの変化を操作者が知ることができたとして、プログラムモジュールのリンク属性を柔軟に変更することができれば、最適な調整を図ることが可能となる。

【0130】また、前記プログラム管理情報はさらに、前記プログラムモジュールそれぞれについて当該プログラムモジュールがプログラム実行過程において呼び出された回数を示す呼出頻度情報を含み、前記プログラムリンク方式はさらに、前記プログラムモジュールそれぞれがプログラム実行過程において呼び出された回数を計測して前記呼出頻度情報を更新する呼出頻度計測手段とを備えることとすることもできる。

【0131】これにより、各プログラムモジュールが呼び出される頻度に応じた制御が可能となる。また、前記即時アンロード可否情報変更手段は、前記プログラムモジュールそれぞれについて、前記呼出頻度情報が所定回数未満を示す場合は当該プログラムモジュールについての前記即時アンロード可否情報を当該プログラムモジュ

ールの実行終了時に当該プログラムモジュールをアンロードする旨を示すようにし、前記呼出頻度情報が所定回数以上を示す場合は当該プログラムモジュールについての前記即時アンロード可否情報を当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードしない旨を示すようにすることとすることもできる。

【0132】これにより、各プログラムモジュールが呼び出される頻度に応じて合理的にメモリの有効利用とプログラムの高速実行の要求のバランスを調整することができる。また、前記プログラムリンク方式はさらに、操作者による前記プログラム管理情報の出力指示を受け付ける第2入力受付手段と、第2入力受付手段から前記出力指示を受け付けた旨の通知を受けたときに前記プログラム管理情報を参照してその内容を出力する出力手段とを備えることとすることもできる。

【0133】これにより、操作者は、現在の各プログラムモジュールのリンク属性等を、出力情報を確認することで正確に把握した上で調整を図ることができるようにする。また、前記即時アンロード可否情報変更手段は、あるプログラムモジュールについての前記即時アンロード可否情報が当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードしない旨を示すようにする場合において、当該プログラムモジュールが前記メモリに格納されていないときは、前記ロード手段を介して当該プログラムモジュールをロードし、前記即時アンロード可否情報が当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードする旨を示すようにする場合において、当該プログラムモジュールが前記メモリに格納されているときは、前記アンロード手段を介して当該プログラムモジュールをアンロードすることとすることもできる。

【0134】これにより、即時的にメモリ使用状況の変更が可能となる。また、前記外部記憶装置はさらに、これが格納する複数の前記プログラムモジュールそれぞれについての前記即時アンロード可否情報を作成するための情報である初期情報を格納し、前記プログラムリンク方式はさらに、あるプログラムの起動時に当該プログラムを構成する複数のプログラムモジュールそれぞれについて前記初期情報に基づき前記即時アンロード可否情報を作成して前記メモリ中へ格納をする即時アンロード可否情報作成手段を備え、プログラム中には、プログラムの実行終了を示す1つのコードと、複数のプログラムモジュールそれぞれを呼び出す旨を示す複数のコードと、複数のプログラムモジュールそれぞれの実行終了を示す複数のコードとが含まれており、前記プログラム実行制御手段はさらに、プログラムの実行終了を示すコードを解読すると当該プログラムを構成する複数のプログラムモジュールのうちメモリに格納されているものすべてを前記アンロード手段を介してアンロードするプログラム

実行終了処理部と、プログラムモジュールを呼び出す旨を示すコードを解読すると当該プログラムモジュールがメモリに格納されていないときには当該プログラムモジュールを前記ロード手段を介してロードするプログラムモジュール呼出処理部とを有することとし、前記プログラムリンク方式はさらに、プログラムを構成する複数のプログラムモジュールそれぞれについて前記初期情報を生成して前記外部記憶装置に格納する初期情報生成手段と、前記複数のプログラムモジュールそれぞれを呼び出す旨のコードと、前記複数のプログラムモジュールそれぞれの実行終了を示すコードとを含む実行可能なコード列からなるプログラムを生成するプログラムコード生成手段と、前記プログラムコード生成手段によって生成したプログラムを複数のプログラムモジュールに分割して、前記プログラムモジュールそれぞれを前記外部記憶装置に格納することとすることもできる。

【0135】これにより、プログラムモジュールのロード、アンロードに関わる制御一切を実現することができる。また、前記初期情報生成手段は、前記プログラムを構成する複数のプログラムモジュールそれぞれについて、当該プログラムモジュールを呼び出す旨のコードの個数を合計した値が所定数以上の場合には、当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードする旨の前記初期情報を生成し、所定数未満の場合には、当該プログラムモジュールの実行終了時に当該プログラムモジュールをアンロードしない旨の前記初期情報を生成することとすることもできる。

【0136】これにより、各プログラムモジュールの実行される頻度は現実にある程度実行した後でなければ情報として利用できないという問題が解決され、予めある程度合理的なメモリ有効利用とプログラム高速実行の調整が図られる。また、本発明に係るメソッドサーチ方法は、動的オブジェクト指向システムにおいてあるオブジェクトからスーパーオブジェクトの有するメソッドを実行するために当該メソッドの所在を検索するメソッドサーチ方式であって、複数のオブジェクトを格納し、メソッド名と当該メソッドの所在を示すリンク情報との組であるメソッド情報を複数格納し得るメモリと、実行すべきメソッド名を受け付けるメソッド名受付手段と、前記メソッド名受付手段により受け付けたメソッド名で示されるメソッドのリンク情報を、前記メモリ中に前記メソッド情報が格納されている場合は、前記メソッド情報から検索し、検索できないとき又は前記メモリ中に前記メソッド情報が格納されていない場合は、前記リンク情報を前記のあるオブジェクトと継承関係にあるオブジェクトのメソッドスロットから検索して取得した上、前記メソッド名と取得した前記リンク情報とをメソッド情報として前記メモリに格納するメソッドサーチ手段と、特定のメソッド情報を前記メモリから削除する削除手段とを備えることを特徴とする。

【0137】これにより、メモリ有効利用の観点からの要求に応え得る動的オブジェクト指向システムが実現できる。即ち、メソッドをサーチした後に蓄積するデータの量が増加した場合等において、このデータをメモリから削除しメモリを開放することにより、メモリの有効利用を図ることができる。ここで、前記メソッドサーチ方式はさらに、削除すべきメソッド情報を特定するためのメソッド情報特定情報を受け付ける削除メソッド受付手段を備え、前記削除手段は、前記削除メソッド受付手段により受け付けたメソッド情報特定情報で示されるメソッド情報を前記メモリから削除することとすることもできる。

【0138】これにより、動的オブジェクト指向システムの利用者は意図どおりにメモリ有効利用とメソッドサーチ高速実行との要求のバランスの調整を図ることができる。また、前記メソッド情報はさらに、それが含むメソッド名に対応するメソッドになされた実行要求の回数に関する情報である頻度値を含み、前記メソッド名受付手段はさらに、受け付けたメソッド名を含むメソッド情報が前記メモリ中に存在する場合にはメソッド情報中の頻度値をカウントアップし、前記削除手段は、所定の要因により起動されるものであり、所定頻度より小さい頻度値を含むメソッド情報をすべて前記メモリから削除することとすることもできる。

【0139】これにより、高頻度で実行されるメソッドについてのリンク情報のみをメモリに残し、ほとんど実行されないメソッドについてのリンク情報については開放することで、最適なメモリ利用が図れる。

【図面の簡単な説明】

【図1】本発明の実施の形態1におけるプログラムリンク方式を実現するためのソースプログラムの例を示す図である。

【図2】ソースプログラム100を参照してプログラムコード作成装置によって作成される実行形式のプログラムコードのイメージを示す図である。

【図3】本発明の実施の形態1におけるプログラムコード作成装置の機能ブロック図である。

【図4】プログラムコード作成装置300の動作を示すフローチャートである。

【図5】本発明の実施の形態1におけるプログラム実行制御装置の機能ブロック図である。

【図6】メモリ管理情報のイメージを示す図である。

【図7】CALL情報のイメージを示す図である。

【図8】実行制御部1100によるプログラム起動の動作を示すフローチャートである。

【図9】MCALL処理部1130の動作を示すフローチャートである。

【図10】RETURN処理部1140の動作を示すフローチャートである。

【図11】EXIT処理部1150の動作を示すフロー

チャートである。

【図12】プログラム実行制御装置1000におけるリンク変更制御動作を示すフローチャートである。

【図13】プログラムモジュールリストのイメージを示す図である。

【図14】実行管理情報記憶部2030に格納されたメモリ管理情報2200のイメージを示す図である。

【図15】リンク変更制御部2110が出力部1020を介して出力するプログラムモジュールリストのイメージを示す図である。

【図16】リンク変更制御部3110の自動リンク変更制御の動作を示すフローチャートである。

【図17】実行管理情報記憶部4030に格納されたメモリ管理情報4200のイメージを示す図である。

【図18】実行制御部4100によるプログラム起動の動作を示すフローチャートである。

【図19】MCALL処理部4130の動作を示すフローチャートである。

【図20】RETURN処理部4140の動作を示すフローチャートである。

【図21】EXIT処理部4150の動作を示すフローチャートである。

【図22】リンク変更制御部4110の動作を示すフローチャートである。

【図23】動的オブジェクト指向システム5000における各オブジェクトの構造のイメージを示す図である。

【図24】メソッドリストの内容のイメージを示す図である。

【図25】メソッド呼出処理の動作を示すフローチャートである。

【図26】動的オブジェクト指向システム5000におけるメソッドリスト削除制御の動作を示すフローチャートである。

【図27】メソッドリストの内容の出力イメージを示す図である。

【図28】削除メッセージを受けたオブジェクトにおけるメソッドリスト削除の動作を示すフローチャートである。

【符号の説明】

100	ソースプログラム
210	プログラムコード
211	モジュール属性定義部
212	実行コード
220	プログラムコード
230	プログラムコード
300	プログラムコード作成装置
310	ソースプログラム解読部
320	モジュール属性定義生成部
330	実行コード生成部
1000、2000、3000、4000	プログラ

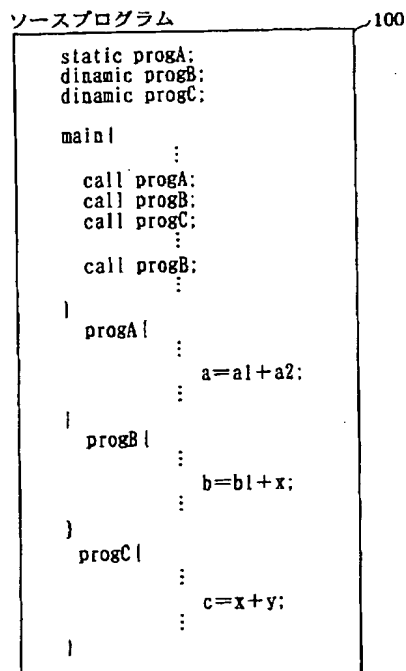
ム実行制御装置

1010 入力受付部
 1020 出力部
 1030、2030、4030 実行管理情報記憶部
 1040 プログラム記憶部
 1050 外部記憶装置
 1100、4100 実行制御部
 1110、2110、3110、4110 リンク変更制御部

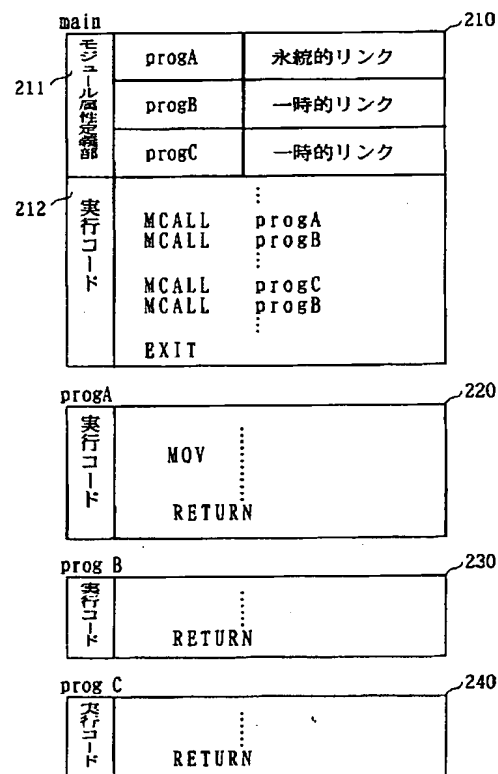
* 1120、4120 実行管理情報作成部
 1130、2130、4130 MCALL処理部
 1140、4140 RETURN処理部
 1150、4150 EXIT処理部
 1200、2200、4200 メモリ管理情報
 1210 CALL情報
 5000 動的オブジェクト指向システム
 5100、5200、5300 オブジェクト
 * 5110、5210、5310 メソッドリスト

【図1】

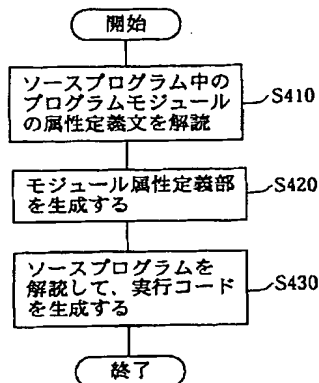
【図2】



【図4】



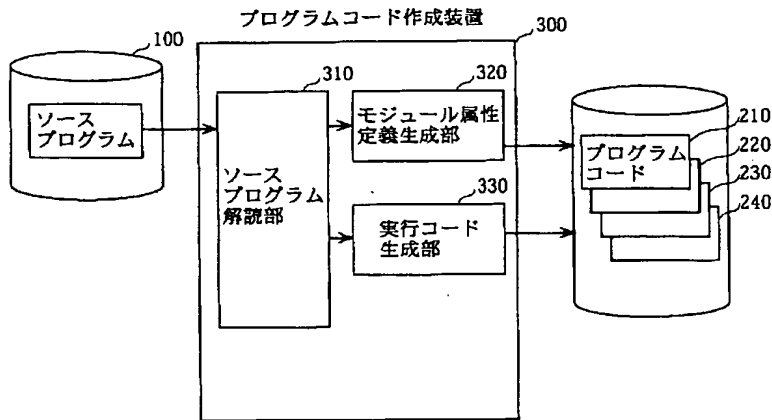
【図6】



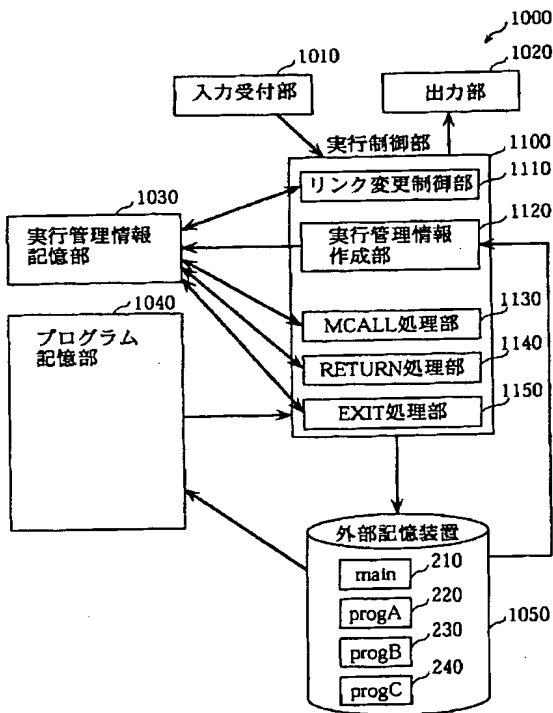
メモリ管理情報

メインプログラム名	アドレス	サイズ		
main	40000	2000		
プログラム モジュール名	リンク属性	アドレス	サイズ	
progA	永続的 リンク	80000	2000	1202
progB	一時的 リンク	—	—	
progC	一時的 リンク	—	—	

【図3】



【図5】

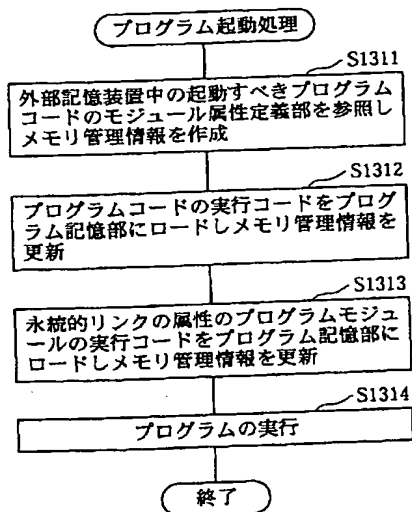


【図7】

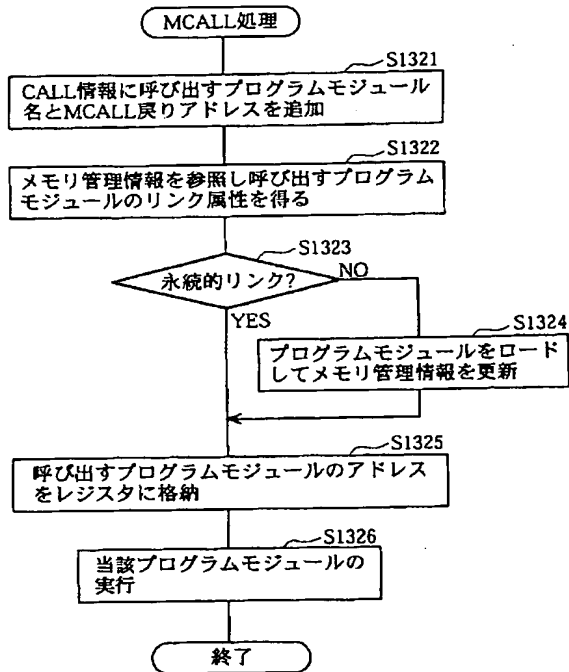
CALL情報 1210

プログラムモジュール名	MCALL戻りアドレス
progA	65000
終端マーク	

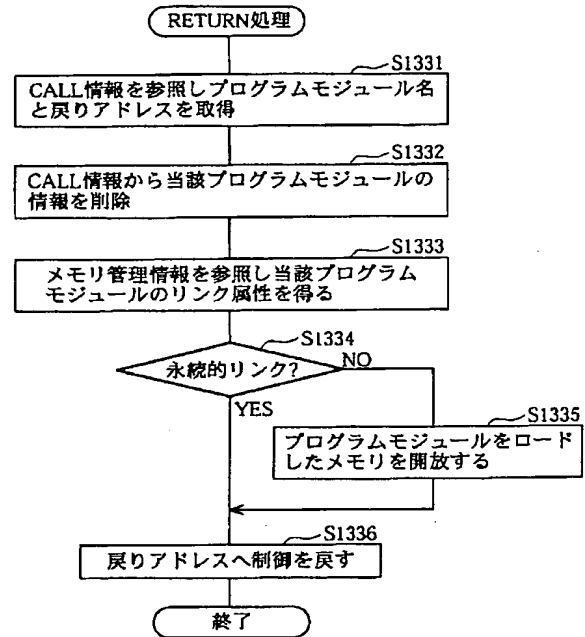
【図8】



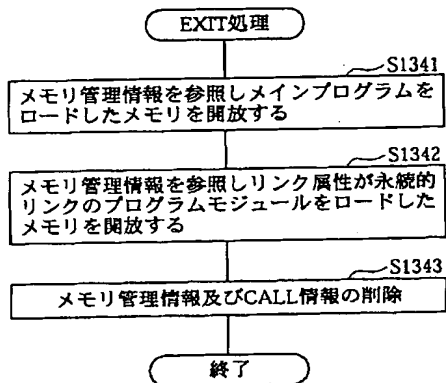
【図9】



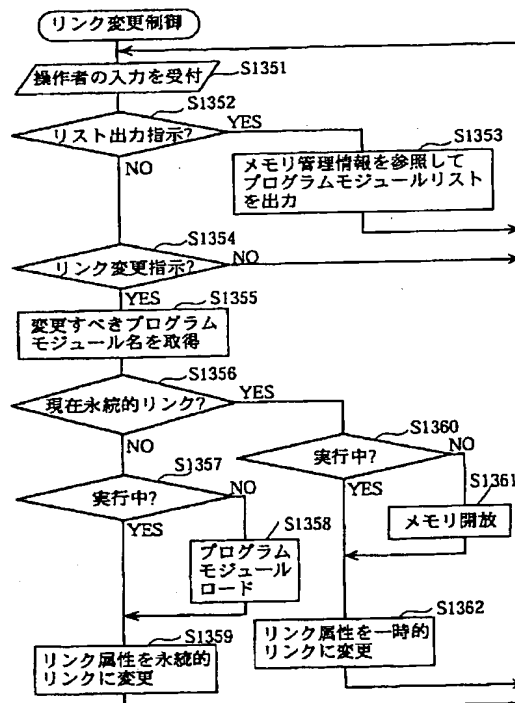
【図10】



【図11】



【図12】



【図13】

プログラムモジュールリスト 1400

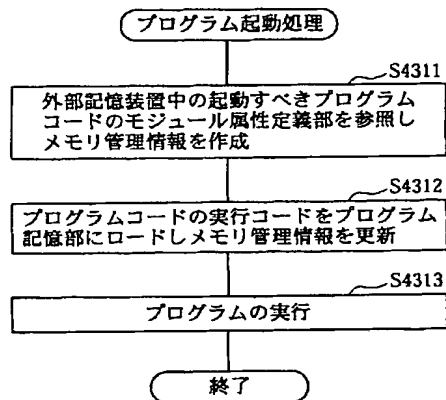
プログラムモジュール名	リンク属性
progA	永続的リンク
progB	一時的リンク
progC	一時的リンク

【図15】

2400
プログラムモジュールリスト

プログラムモジュール名	リンク属性	呼出頻度
progA	永続的リンク	10
progB	一時的リンク	2
progC	一時的リンク	1

【図18】



【図14】

2200

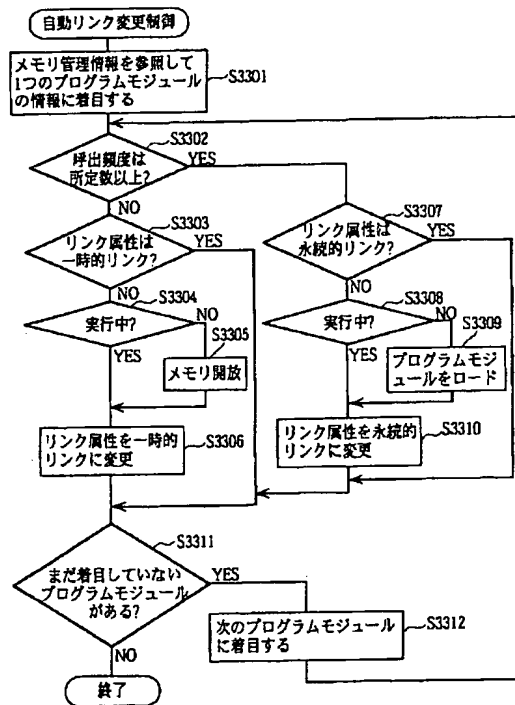
メインプログラム名	アドレス	サイズ
main	40000	2000

2201

プログラムモジュール名	リンク属性	アドレス	サイズ	呼出頻度
progA	永続的リンク	80000	2000	10
progB	一時的リンク	—	—	6
progC	一時的リンク	—	—	1

2202

【図16】



【図24】

5400

メソッド名	ポインタ	呼出頻度
MethodA	80000	10
MethodB	88000	1

【図17】

4200

メインプログラム名	アドレス	サイズ
main	40000	2000

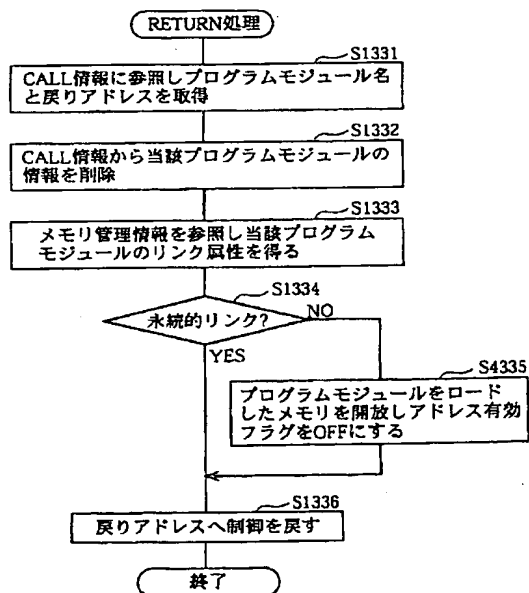
プログラム モジュール名	リンク属性	アドレス	サイズ	アドレス 有効フラグ
progA	永続的 リンク	—	—	OFF
progB	一時的 リンク	—	—	OFF
progC	一時的 リンク	—	—	OFF

【図27】

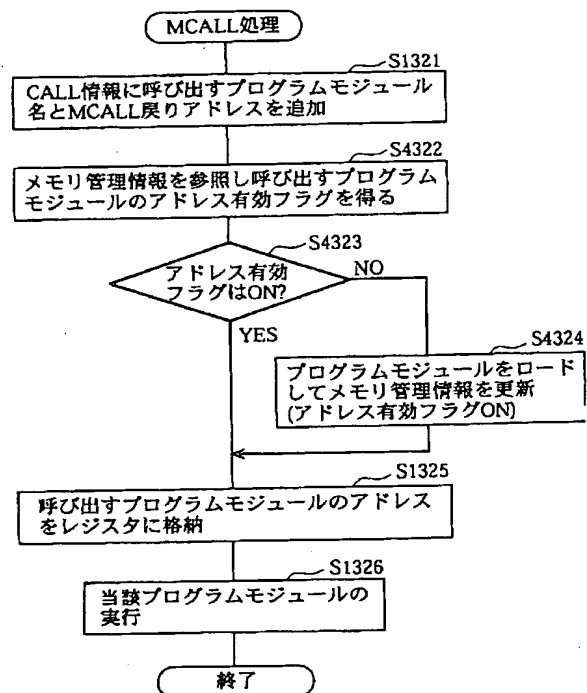
5600

オブジェクト名	メソッド名	呼出頻度
ObjectC	MethodA	10
ObjectC	MethodB	1

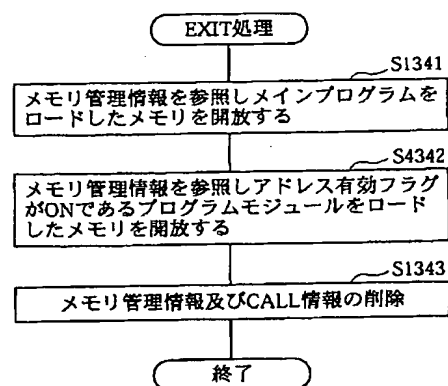
【図20】



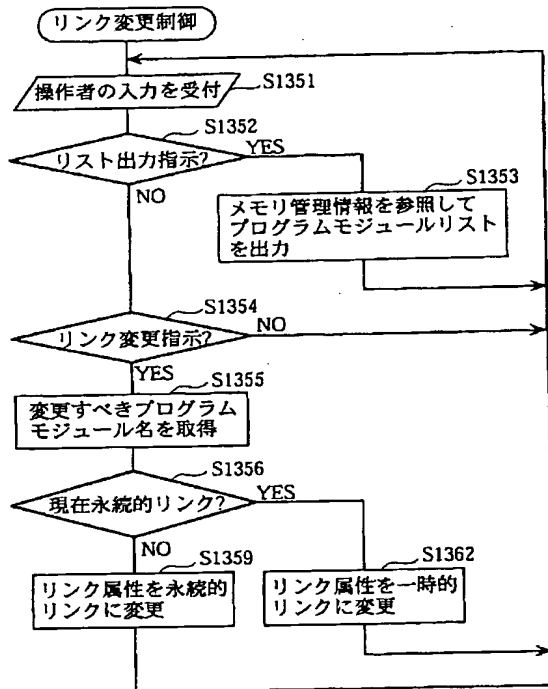
【図19】



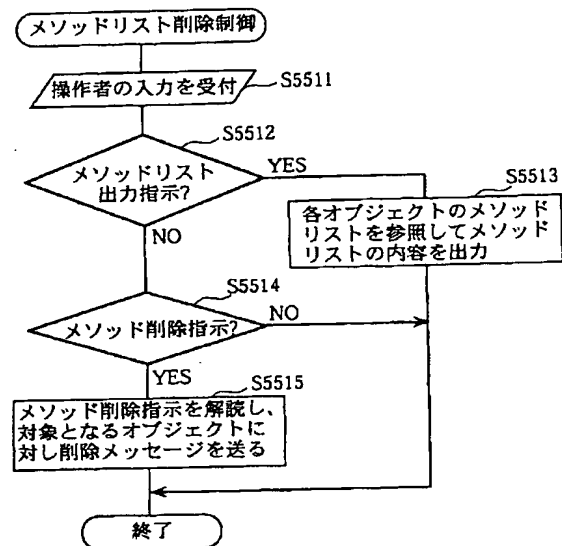
【図21】



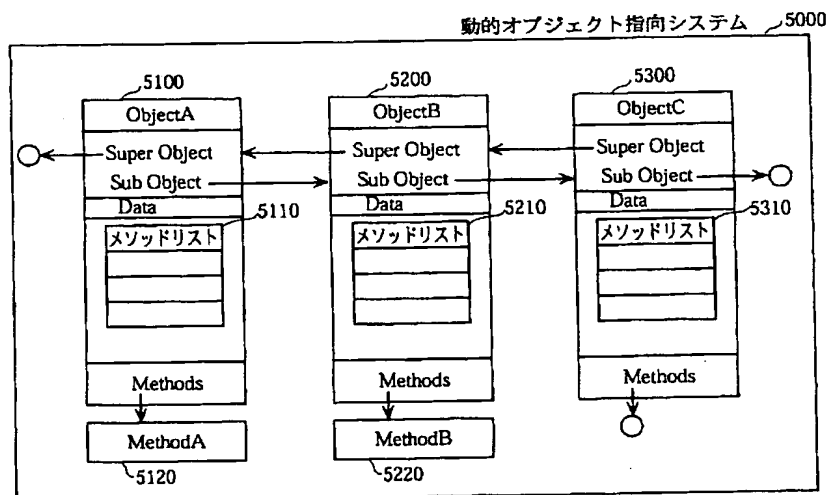
【図22】



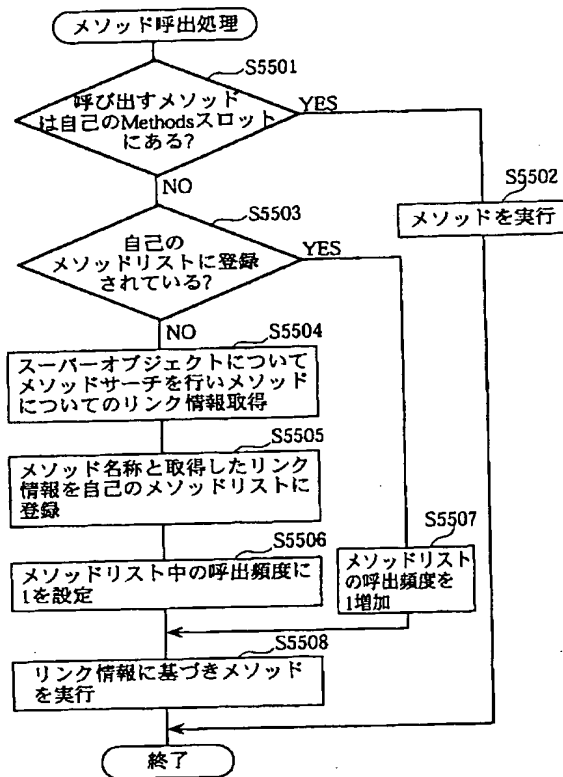
【図26】



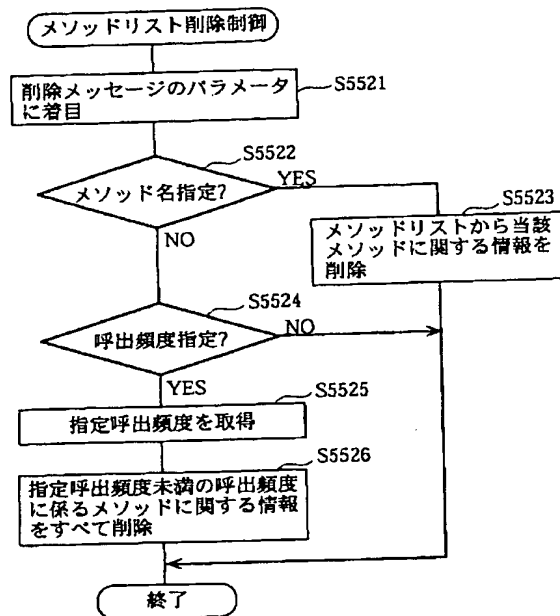
【図23】



【図25】



【図28】



フロントページの続き

(72)発明者 和氣 裕之

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

(72)発明者 石川 亮

大阪府門真市大字門真1006番地 松下電器
産業株式会社内